



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**PSANÍ NA POČÍTAČI POMOCÍ MOZKOVÝCH SIGNÁLŮ**

TYPING USING BRAIN SIGNALS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**LUKÁŠ WAGNER**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**JAN TINKA, Ing.**

**BRNO 2021**

## Zadání bakalářské práce



Student: **Wagner Lukáš**  
Program: Informační technologie  
Název: **Psaní na počítači pomocí mozkových signálů**  
**Typing Using Brain Signals**  
Kategorie: Bioinformatika  
Zadání:

1. Nastudujte problematiku komunikace mezi mozkiem a počítačem (BCI) využívající elektroencefalografu (EEG). Seznamte se se zařízením OpenBCI. Prostudujte metody strojového učení pro rozpoznávání EEG signálů.
2. Navrhněte vlastní implementaci rozpoznávání EEG signálů a nad tím postavenou aplikaci, do které bude možné pomocí EEG psát.
3. Navrženou aplikaci implementujte.
4. Ověřte správnost fungování vaší aplikace na lidech, pokud to umožní epidemiologická situace. Navrhněte možné pokračování práce. Klíčové vlastnosti práce prezentujte formou krátkého videa a plakátu.

### Literatura:

- Wolpaw, J. a Wolpaw, E. W., eds. Brain-computer interfaces: principles and practice. Oxford University Press, 2012. ISBN 978-0-19-538885-5.
- Siuly, S., Li, Y. a Zhang, Y. EEG Signal Analysis and Classification. Springer International Publishing, 2016. Health Information Science. ISBN 978-3-319-47652-0. Dostupné z: <http://link.springer.com/10.1007/978-3-319-47653-7>.
- Cohen, M. X. Analyzing neural time series data: theory and practice. The MIT Press, 2014. Issues in clinical and cognitive neuropsychology. ISBN 978-0-262-01987-3.
- Luck, S. J. An introduction to the event-related potential technique. Second edition. The MIT Press, 2014. ISBN 978-0-262-52585-5.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a částečně 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Tinka Jan, Ing.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

## Abstrakt

Tato práce se zabývá implementací rozhraní mezi mozkem a počítačem v jazyce Python umožňující psát a komunikovat s použitím zařízení EEG. Práce studuje a hodnotí dosavadní technologie rozhraní mezi mozkem a počítačem pro použití za tímto účelem. Dále se práce zabývá použitím strojového učení, které se v technologii používá, zejména neuronovými sítěmi, jež se prokázaly být jednou z nejpřesnějších metod klasifikace signálu EEG. Následně jsou navrženy a implementovány 3 různé systémy založené na různých paradigmatech vyvolávání změny potenciálu EEG vizuální cestou. Tyto systémy byly na závěr otestovány s různými přístupy ke zpracování signálu. Bohužel žádný ze systémů neuspěl v komunikaci cílových písmen.

## Abstract

This bachelor thesis focusses on the implementation of a brain-computer interface, programmed in Python language, that would enable to communicate using EEG. The thesis investigates and evaluates existing brain-computer interface technologies for this purpose. The thesis also explores the use of machine learning applied to the technology, in particular neural networks, which have proven to be one of the most accurate methods of EEG signal processing. Following that, 3 different systems are proposed and implemented, each on different paradigm of visually evoking EEG potential changes. These systems were tested with different signal classification approaches. Unfortunately, none of the systems proved to be usefull in communication.

## Klíčová slova

BCI, EEG, rozhraní mezi mozkem a počítačem, mozek, elektroencefalograf, VEP, SSVEP, f-VEP, P300, c-VEP, vizuálně evokované, Python, OpenBCI, klávesnice, CNN, konvoluční neuronová síť

## Keywords

BCI, EEG, brain-computer, brain, electroencephalograph, VEP, SSVEP, f-VEP, P300, c-VEP, visually evoked, Python, OpenBCI, speller, CNN, convolutional neural network

## Citace

WAGNER, Lukáš. *Psaní na počítači pomocí mozkových signálů*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Jan Tinka, Ing.

# Psaní na počítači pomocí mozkových signálů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Jana Tinky. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Lukáš Wagner  
11. května 2021

## Poděkování

Chtěl bych poděkovat panu Tinkovi za jeho vedení a odbornou pomoc a také své rodině, která si se mnou dělala starosti a pomáhala při kontrole práce.

# Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Úvod</b>   | <b>2</b>  |
| <b>2</b> | <b>Současné technologie rozhraní mezi mozkem a počítačem</b>            | <b>3</b>  |
| 2.1      | Elektroencefalogram . . . . .   | 3         |
| 2.2      | Typy rozhraní mezi mozkem a počítačem podle mentální aktivity . . . . . | 5         |
| 2.3      | Použití umělé inteligence . . . . .                                     | 9         |
| <b>3</b> | <b>Návrh a implementace aplikace</b>                                    | <b>12</b> |
| 3.1      | Návrh . . . . .   | 12        |
| 3.2      | Knihovny jazyka Python . . . . .  | 13        |
| 3.3      | Uživatelské rozhraní systému . . . . .                                  | 14        |
| 3.4      | Detaily implementace . . . . .  | 16        |
| 3.5      | Předzpracování a generování dat . . . . .                               | 18        |
| 3.6      | Vlastní dataset . . . . .   | 20        |
| 3.7      | Využití strojové učení . . . . .  | 20        |
| <b>4</b> | <b>Testování systému na subjektech</b>                                  | <b>26</b> |
| 4.1      | Výsledky první iterace systému c-VEP . . . . .                          | 26        |
| 4.2      | Výsledky druhé iterace systému c-VEP . . . . .                          | 28        |
| 4.3      | Výsledky systému P300 . . . . .   | 29        |
| 4.4      | Výsledky systému SSVEP . . . . .  | 30        |
| 4.5      | Shrnutí výsledků a diskuze . . . . .                                    | 31        |
| <b>5</b> | <b>Závěr</b>  | <b>34</b> |
|          | <b>Literatura</b>   | <b>36</b> |

# Kapitola 1

## Úvod

Brain-Computer Interface (BCI, Rozhraní mezi mozkem a počítačem) je technologie umožňující přímé napojení mozkové aktivity jako vstup pro počítač. Tato technologie umožňuje lidem trpícím ochrnutím končetin, například kvůli amyotrofické laterální skleróze, úrazu páteře nebo dětské obrně, komunikovat se svým okolím nebo ovládat přístroje jako elektrický invalidní vozík nebo rozsvěcovat světla a otvírat dveře v prostředí chytré domácnosti, čímž navyšuje jejich samostatnost a kvalitu života.

První Brain-Computer Interface byly založeny na schopnosti člověka naučit se ovládat svou mozkovou aktivitu natolik, aby byl počítač schopný rozeznat jednotlivé zamýšlené akce. Na poli BCI se uplatnilo použití strojového učení a v současné době populárního hlubokého učení, čímž se urychlila doba nasazení systému na konkrétním uživateli a přenesla se zátěž z uživatele na systém. Za účelem dalšího urychlení se také začalo používat i jiných mozkových signálů, mimo jiné signálů vyvolaných v reakci na nějaký podnět, nejčastěji vizuální. Existuje varianta BCI zaváděná přímo do mozku, tzv. invazivní, a varianta zachytávající signály z vnějšku, tzv. neinvazivní. Výhodou invazivní metody je nízký šum signálu, elektrody jsou zaváděny přímo na požadované místo, ovšem vyžadují chirurgický zákrok. Invazivní metody současně stojí před etickým problémem své bezpečnosti a technologickým problémem dlouhodobého nasazení, často totiž dochází k imunitní reakci, která oddělí elektrody od povrchu mozku (cysty), zatímco neinvazivní metody již našly komerční uplatnění například v herním průmyslu.

Cílem práce je návrh a implementace BCI založeném na vizuálním podnětu, umožňující subjektu psát. Programová část je napsána v jazyce Python pro svou jednoduchost implementace a možnost použití vysokoúrovňových knihoven. Grafické rozhraní předpokládá použití standardního LED monitoru s obnovovací frekvencí 60 Hz a rozlišením 1920 na 1080 pixelů. Pro snímání mozkových signálů je použita neinvazivní metoda s EEG zařízením OpenBCI.

Ve druhé kapitole práce jsou rozebrány současné technologie BCI, jako jsou mentální aktivity používané pro generaci signálu a jejich vhodnost použití. Použití strojového učení i hlubokého učení v oblasti klasifikace signálu elektroencefalografu a popisu tohoto signálu a jeho časté nedostatky. Třetí kapitola obsahuje původní návrh aplikace a popis finální implementované podoby. Implementace je rozdělena do tří částí: Grafické uživatelské rozhraní, předzpracování signálu a strojové učení. Také obsahuje zobrazení detailu neuronové sítě použité pro klasifikaci signálu. V kapitole čtvrté je navrženo testování implementovaných systémů a analyzován výsledek přesnosti klasifikace, rychlosti nasazení a přenosové rychlosti. na skutečných subjektech.

## Kapitola 2

# Současné technologie rozhraní mezi mozkiem a počítačem

Cílem obecného rozhraní mezi mozkiem a počítačem je zajistit komunikaci mezi počítačem a člověkem za pomoci snímání signálů generovaných mozkiem daného člověka. Pokud vynecháme účel (elektrický vozík, klávesnice, hra, atd.), tak obecný BCI lze kategorizovat podle čtyř částí, v nichž se jednotlivé implementace mohou lišit: *generace signálu*, *výběr příznaků signálu*, *klasifikace příznaků* a *typ mentální aktivity* [4, 24].

Pro generaci signálu se nejčastěji používá elektroencefalograf, ale lze využít i jiných metod, jako například funkční blízká infračervená spektroskopie [7], na které lze sledovat hladiny kyslíku v krvi v mozku, nebo funkční transkraniální dopplerovská sonografie na ultrazvuku [10], která měří proudění krve v mozku.

### 2.1 Elektroencefalogram

Elektroencefalogram je signál měřený elektroencefalografem. Elektroencefalograf (EEG) je zařízení, které snímá drobné rozdíly potenciálů vyvolaných mozkovou aktivitou. EEG je tvořeno elektrodami, jež měří potenciál v místě kontaktu. Často se používá elektroda referenční, která se umísťuje na ucho, kde není očekávána žádná změna potenciálu. Pokud se takováto elektroda nepoužije, je reference získána jiným způsobem. Vůči této elektrodě se měří napětí z potenciálů ostatních elektrod. Často se také používá elektroda kontrolní, která se umísťuje na ucho druhé, tato elektroda by v ideálním případě měla mít nulové napětí vůči elektrodě referenční. Existují elektrody invazivní (zaváděné do mozku), poloinvazivní a neinvazivní, kterými se tato práce dále zabývá. Neinvazivní elektrody se dotýkají povrchu skalpu a dále se dělí na suché a plovoucí. Plovoucí elektrody využívají vodivé pasty pro zaručení kontaktu a sběru signálu. Lepivost této vodivé pasty je velkou nepříjemností pro lidi s vlasy. Suché elektrody mohou být také nepříjemné, protože využívají ostrých hran, aby se dostaly skrz vlasy a měly přímý kontakt s kůží.

Důvodem šumu elektrod je malý poměr síly mozkových signálů na signály cizí. Změny potenciálů v mozkové kůře jsou relativně malé (v řádu  $mV$ ), které jsou na skalpu ještě slaběji měřitelné (v řádu  $\mu V$ ). Za účelem sběru těchto signálů musí být elektrody dostatečně citlivé a jsou ovlivněny řadou signálů silnějších. Jelikož elektrody snímají veškeré změny potenciálu na skalpu, výsledný signál může obsahovat i elektrický signál svalových nervů, jako například mrkání. Výrazný šum také způsobuje elektrická síť používaná okolními zařízeními. Elektrická síť v Evropě kmitá na frekvenci 50 Hz a v Severní Americe 60 Hz. Elektromag-

netická indukce způsobuje pozorovatelný šum v signálu na zmíněných frekvencích. Proto bývá relevantní rušivá frekvence kompletně odfiltrována z dat.

Oscilační mozkové signály (vlny) se běžně dělí do 5 skupin (frekvenčních pásem), jednotlivým vlnám se přidělují činnosti které je vyvolávají [1]. V oblasti BCI se běžně používají frekvence od 0 do 40 Hz. Nejlépe se aktivita pozoruje ve vlnách alfa, které mají největší amplitudu z vln vyvolaných aktivní činností. Všeobecně u mozkových vln platí, že čím větší frekvence, tím menší amplituda [14]. Na signál se tedy aplikuje kombinace dolní a horní propusti, vzniká tak filtr v podobě pásmové propusti a nechtěné frekvence se odfiltrují.

| Název | Frekvence | Činnost                     |
|-------|-----------|-----------------------------|
| Delta | 0.5–4 Hz  | Spánek                      |
| Théta | 4–8 Hz    | Meditace, relaxace          |
| Alfa  | 8–12 Hz   | Relaxace, pasivní pozornost |
| Beta  | 12–35 Hz  | Aktivita, aktivní pozornost |
| Gama  | >35 Hz    | Koncentrace                 |

Tabulka 2.1: Mozkové vlny a činnosti je vyvolávající

Evokovaný potenciál nebo potenciál v návaznosti na událost (ERP) je odlišitelná změna v přirozeném mozkovém signálu následující po prezentaci stimulu. Amplituda evokovaného potenciálu bývá relativně malá a vyžaduje průměrování signálů. ERP je v signálu časově svázan s prezentací stimulu, zatímco šum je náhodného charakteru a průměrováním přes více prezentací stimulu jej tedy lze částečně odstranit [1].

## Zařízení OpenBCI

Zařízení OpenBCI je projekt s otevřeným kódem, je tedy možné zadarmo stáhnout software i plány obvodů nebo 3D modely pro tisk. Pokud člověk nemá možnost tisku helmy nebo vytvoření obvodů, lze na oficiálních stránkách<sup>1</sup> zakoupit celý set. Základní verze obvodu Cyton obsahuje 8 elektrod, ale jejich počet lze rozšířit o dalších 8 rozšířením Daisy. V projektu byl použit obvod Cyton s rozšířením Daisy.

Headset zařízení je navržen tak, aby obsahoval nutné obvody i baterii, a umožňuje připojit až 16 suchých šroubovacích elektrod na některá z 35 různých míst, podle celosvětově uznávaného systému 10–20, zobrazeného na obrázku 2.1. Do zbylých míst lze umístit plochý šroub pro lepší uchycení na hlavě.

Zařízení komunikuje s počítačem bezdrátově za použití technologie *bluetooth low energy* (BLE). Součástí balíčku OpenBCI je také *USB Dongle*, který umožňuje bluetooth spojení s headsetem a není tedy zapotřebí, aby bluetooth podporoval sám počítač.

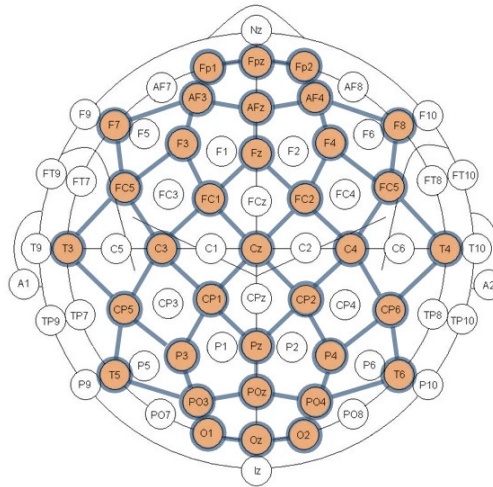
## Zpracování signálu

Před předáním signálu klasifikačním algoritmům, které mají na starosti finální určení sdělované informace, často dochází k extrakci příznaků signálu. Příznaky reprezentují podstatné informace obsažené v signálu v kompaktní podobě. Může se jednat například o zastoupení určitých frekvencí nebo největší rozdíl mezi špičkou a propadem (v amplitudě). Dochází tak k redukci velikosti dat a potlačení šumu. Pro výběr vhodných příznaků se signál často transformuje.

<sup>1</sup><https://openbci.com/>



## Ultracortex Mark IV Node Locations (35 total)



Based on the internationally accepted **10-20 System**  
for electrode placement in the context of EEG research

Obrázek 2.1: Umístění elektrod na headsetu OpenBCI. Převzato, z oficiálních stránek <https://openbci.com/>

Jednou z možných transformací signálu je *Fourierova transformace*. Fourierova transformace převádí signál z domény časové do domény frekvenční. Frekvenční doména znázorňuje frekvence obsažené v signálu, jedná se tedy o transformaci často používanou pro f-VEP, jak dále zmíním v sekci 2.2.

Další transformací signálu je *vlnková transformace*, která převádí signál do domény časově frekvenční. V této doméně lze vypořizovat existenci a trvání frekvenčních událostí, je tedy vhodná pro sledování potenciálů vyvolaných neočekávaným vjemem, jak dále zmíním v sekci 2.2.

## 2.2 Typy rozhraní mezi mozkiem a počítačem podle mentální aktivity

Podle Ali Bashashati a kol. [4] lze BCI rozdělit podle mentální aktivity do 7 kategorií.

- *senzomotorické aktivity*
- *pomalé potenciály v mozkové kůře*
- *reakce na nečekávaný podnět*
- *vizuálně evokovaný potenciál*
- *odezva na mentální úkol*
- *aktivita nervových buněk*

- *kombinace aktivit*

Dále jsou zmíněny pouze kategorie, které osobně považuji za důležité ve vývoji BCI a pro tento projekt.

## Pomalé potenciály v mozkové kůře

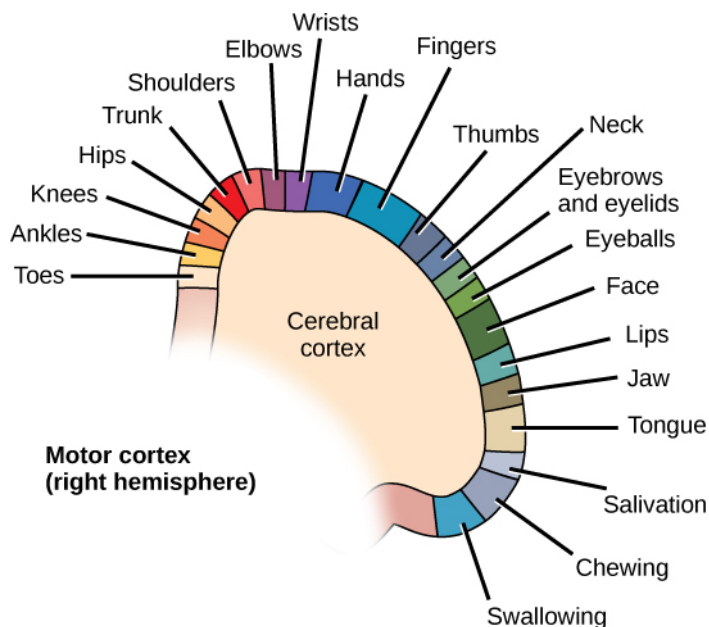
Jednou z prvních metod Brain-Computer Interface je využití pomalých potenciálů kůry. Jedná se o změny v časovém rozmezí od 0.5 do 10 sekund. Negativní změna v potenciálu se spojuje s pohybem a pozitivní se sníženou aktivací v kůře [23]. Metoda je zajímavá pro svůj přímý „překlad myšlenek“ bez pomoci externích stimulů nebo specifických úkolů. Může se tedy na první pohled jevit jako krok směrem ke čtení myšlenek.

Člověk, který se nikdy s metodou nesetkal se musí naučit „ovládat své myšlenky“ natolik, aby byl systém schopný je klasifikovat. Její nasazení je tedy velmi časově náročné.

## Senzomotorické aktivity

Senzomotorické aktivity obsahují kromě aktivit pohybových také pouhou *představu pohybu*. Úkolem subjektu je představit si, jak pohybuje končetinami, aniž by jimi skutečně pohyboval. Tato akce vyvolává měřitelné potenciály v mozkové kůře a je založena na události přípravy k pohybu. Pokud člověk záměrně pohne končetinou, lze již přibližně 2 sekundy předem naměřit mozkovou aktivitu rozhodování vedoucí k pohybu.

Za pohyb těla zodpovídá *pohybová kůra mozku*, která je umístěná na čelním laloku mozku. Jedná se o úzký pás mozkové kůry, který se dále dělí na jednotlivé podoblasti. Každá z těchto podoblastí zodpovídá za některé svalové skupiny, rozdělení podoblastí je znázorněno na obrázku 2.2.



Obrázek 2.2: Rozdělení pohybové kůry mozkové

Převzato z CNX OpenStax - [http://cnx.org/contents/GFy\\_h8cu@10.53:rZudN6XP@2/Introduction](http://cnx.org/contents/GFy_h8cu@10.53:rZudN6XP@2/Introduction), <https://commons.wikimedia.org/w/index.php?curid=49934991>

Metoda je nevýhodná z pohledu komunikace kvůli nízkému množství kategorizovatelných „myšlenek“, ještě více omezených velkým poměrem šumu na signál u EEG. Nasazení je složité, jelikož metoda vyžaduje krom strojového učení také trénink ze strany uživatele. Přizpůsobení se je ovšem rychlejší oproti naučení se ovládání pomalých potenciálů.

## Reakce na nečekaný podnět

Reakce na nečekaný podnět, konkrétně její část nazývaná P300, která se tak jmenuje pro svou charakteristickou vlastnost velkého pozorovatelného výkyvu v EEG 300 ms po výskytu podnětu. Pokud se jako podnět využívá zrakový vjem, lze P300 také klasifikovat jako Časově modulovaný vizuálně evokovaný potenciál. Aktivita spočívá většinou ve vyčkávání na objevení se subjektem očekávaného signálu. Například lze subjektu přehrávat různé permutace znaků abecedy, zatímco subjekt má za úkol počítat výskyty jím chtěného písmene (P300 je špatně detekovatelné, pokud subjekt vjem očekává v přesném čase). Počítání výskytů je jedna z metod zajištění soustředění se subjektu.

Signál vyvolaný nečekaným podnětem je měřitelný po celém mozku, ovšem nejlépe jej lze snímat na temenním laloku.

Výhodou metody je, že není zapotřebí učení ze strany uživatele a strojové učení není nutností, je tedy dobře nasaditelná. Ovšem její komunikační rychlost zpomaluje nutnost průměrování výsledku, nelze totiž s jistotou říci, že P300 je skutečně obsažené v signálu a nejedná se o šum nebo, v případě tabulky písmen, lze také naměřit slabší falešné P300 pro stimuly sousední cílovému stimulu. Metoda dosahuje přenosové rychlosti až 30 *bit/min* [5].

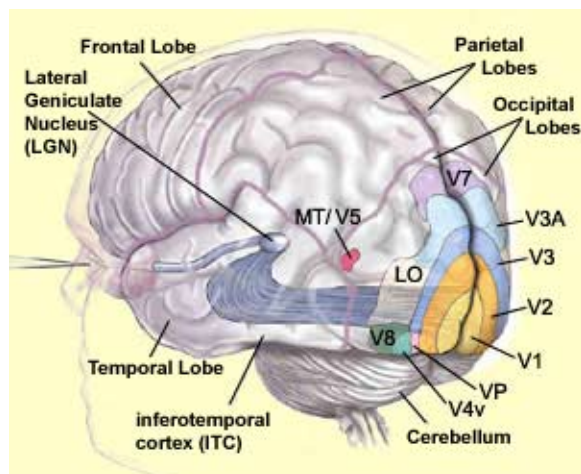
## Frekvenčně modulovaný vizuálně evokovaný potenciál

Zkráceně f-VEP, také nazýván *Steady-State Visually Evoked Potential* (SSVEP). Jedná se o synchronizaci aktivity neuronů s frekvencí stimulu, pokud tedy subjekt pozoruje písmeno blikající na dané frekvenci, lze tuto frekvenci nebo její harmonické vypočítat v EEG zejména týlního laloku subjektu.

I když lze f-VEP naměřit ve všech frekvencích od 1 Hz do nejméně 90 Hz [9], nejlépe měřitelné jsou frekvence v okolí alfa mozkových vln (8–12 Hz) [17]. Problém také nastává při klasifikaci dvou navzájem prvních harmonických frekvencí. Použití klasického monitoru o obnovovací frekvenci 60 Hz zanáší do systému šum a limituje spolehlivost vykreslování na určité frekvence podporované obnovovací frekvencí, konkrétně 7.5 Hz, 8.57 Hz, 10 Hz, 12 Hz, 15 Hz, respektive 8, 7, 6, 5, 4 snímků na jednu periodu [17]. Vzhledem k tomuto omezení se při použití f-VEP využívá k agregaci písmen do skupin, dochází tedy prvně k výběru skupiny obsahující požadovaný znak a teprve posléze k výběru znaku ze skupiny.

Stimuly mohou problikávat samy o sobě („single graphic“), nebo mohou invertovat barvu se svým okolím („pattern reversal“). Ovšem blikání stimulů na uvedených frekvencích je často spojováno s námahou na oči. Za účelem snížení této námahy bylo navrženo více různých přístupů k f-VEP, jako zrychlení frekvencí, nahrazení blikání za plynulý pohyb nebo amplitudová modulace [6].

U všech vizuálně evokovaných potenciálů platí, že se nejlépe rozpoznávají v týlním mozkovém laloku, kde se také nachází zrakové centrum, které tyto vjemy zpracovává. Umístění zrakových center je znázorněno na obrázku 2.3.



Obrázek 2.3: Zobrazení zrakových center v mozku.

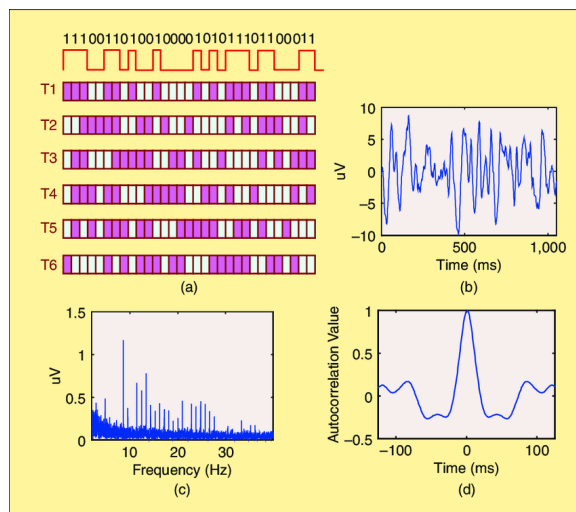
Převzato z <https://visionhelp.wordpress.com/2018/07/04/the-visual-centers-of-the-brain/>

Výhodou metody je opět nepotřebnost lidského a strojového učení. Také pro svou relativní rychlost až  $60 \text{ bit/min}$  [5] a jednoduchost implementace se jedná o jednu z nejpoužívanějších metod.

### Kódově modulovaný vizuálně evokovaný potenciál

Zkráceně c-VEP, je metoda na poli BCI poměrně mladá. I když ji navrhl Erich E. Sutter již v roce 1984, nebyla metoda dlouho poté podrobně prozkoumána. Využívá pseudo náhodných sekvencí ke generování blikání stimulu, např. *m-sekvence* založené na posuvných registrech s lineární zpětnou vazbou. M-sekvence patří k nejpoužívanějším pseudonáhodným kódům v metodě c-VEP [5]. Všechny stimuly blikají podle stejné posloupnosti ovšem různě posunuté v čase, toto je ilustrováno na obrázku 2.4. Pseudonáhodné posloupnosti jako m-sekvence se používají, protože jejich autokorelační funkce je podobná signálu jediného pulzu, který dosahuje hodnoty 1. Těto vlastnosti se využívá při klasifikaci signálu. Uvažujeme čistý pulz vytvořený autokorelační funkcí a nulovou reakční dobu subjektu, pokud signál nalezený v EEG koreluje s některým ze stimulů na obrazovce za generace 1, pak jsou signály shodné a je velmi pravděpodobné, že tento stimul je požadovaným cílem subjektu. Se všemi ostatními stimuly poté signál koreluje za vzniku 0. Tento přístup je ovšem velmi naivní, protože sekvence se přesně nepromítne do mozkové aktivity. Spíše se objeví pouze hrany sekvence jako vrcholky v signálu. Proto se v trénovací fázi systému vytváří vzor promítnutého signálu průměrováním z mnoha pokusů na stejně posunuté sekvenci. Tento vzor se při klasifikaci posune pro každé použité posunutí sekvence a provede se jeho korelace se signálem.

Jednou z nevýhod metody je nutnost trénování klasifikace systému před použitím. Byly provedeny experimenty umožňující trénování klasifikátoru během aktivního používání, ovšem tento přístup není ze začátku schopen klasifikovat prvních několik znaků a klasifikuje je zpětně [22]. Jinak je metoda pro použití jako klávesnice jedna z nejvhodnějších. V příkladech uvedených v [5] metoda dosahovala rychlostí přenosu informace  $100+ \text{ bit/min}$  [5], a to díky své možnosti zobrazení velkého množství stimulů, prakticky jednoho na jeden stav sekvence, počet stimulů tedy záleží na délce sekvence, se zvětšující se délkou sekvence ovšem



Obrázek 2.4: Stimuly a signál vyvolaný c-VEP a) sekvence stimulů v jednom cyklu, lze vidět posunutí b) vyvolaný signál c) spektrum signálu d) autokorelační funkce signálu  
Převzato z [5]

klesá rychlost rozeznání jednotlivých vjemů [5]. Jak bylo výše zmíněno, vizuálně evokované potenciály se snímají z týlního mozkového laloku, kde se nachází zraková centra (Obrázek 2.3).

## 2.3 Použití umělé inteligence

V problematice BCI se uplatnilo použití strojového učení, které je rychlejší než původně používaná metoda *biofeedback*, která spoléhala na lidskou přizpůsobivost potřebným vstupům. Cílem metody *biofeedback* je naučit se ovládat jednoduchou hru, která reaguje na určité mozkové vlny. Takto je člověk nucen „učit se“ soustředit a i v dnešní době má metoda použití jako terapie pro lidi trpící poruchami pozornosti nebo pro lidi se spánkovými problémy. Cílem použití strojového učení je ovšem naopak „přizpůsobit“ systém osobním potřebám člověka [13].

Při vývoji BCI se nejčastěji používá *učení s učitelem* (*supervised learning*). Systému je poskytnut vstupní signál spolu s jeho cílem. V praxi to znamená, že uživatel „kalibruje“ program tím, že se pokouší opsat předem daný text. Za použití *hlubokého učení* je možné extrahovat přímo ze signálu jeho příznaky, popřípadě jednodušším strojovým učením vytvořit klasifikace příznaků vypočítaných předem určeným způsobem. Nejpoužívanější typ *hlubokého učení* pro komunikaci je *konvoluční neuronová síť* [24, 18], méně také *rekurentní neuronová síť* nebo „*restricted Boltzmann machine*“ [18]. A mezi používané algoritmy klasifikace strojovým učení patří metoda podpůrných vektorů, lineární diskriminační analýza nebo klasifikace velkými rozestupy (*large margin classification*) [13]. Hluboké učení umožňuje použít na vstupu nezpracovaný signál a k identifikaci příznaků dojde učením na skrytých vrstvách neuronové sítě [24, 8].

Ale některé metody, jako například metoda kódově modulovaného vizuálně evokovaného potenciálu, využívají i metody *učení bez učitele*, při kterém je strojovému učení poskytnut signál k trénování ovšem bez cíle. Předpokládá se, že si strojové učení vytvoří něco jako „průměrný signál“ a jeho úkolem je následně určit, jak moc se klasifikovaný signál odlišuje

od naučeného „průměru“. Pokud je použit tento přístup, je často zapotřebí trénovat větší množství modelů strojového učení a klasifikaci signálu provést na bázi nejmenší vzdálenosti od naučených „průměrných signálů“. [21]

Výkon strojového učení závisí krom zvoleného algoritmu také na správném výběru příznaků. Jak uvádí Lotte a kol. v [12], v různých studiích bylo použito množství rozdílných příznaků založených na amplitudách, „band power“, výkonovém spektru, časově-frekvenční doméně (spektrogram) a také auto-regresivní příznaky. V případě EEG signálu příznaky trpí špatným odstupem signálu od šumu, velkým množstvím příznaků na popis signálu, změnou signálu v čase, relativně malým množstvím trénovacích dat a u některých typů příznaků zachováním informace o čase.

Za účelem určení změny signálu v čase byly navrženy tři přístupy: spojení příznaků z více časových segmentů a následné klasifikace, spojení klasifikací vzniklých z jejich příznaků z více časových segmentů a dynamická klasifikace [12]. Nejvíce se využívá prvního výše zmíněného přístupu, který ale zapříčiňuje výše zmíněný problém – nárůst příznaků [12].

## Convolutional Neural Network

Konvoluční neuronová síť (CNN) je jeden z druhů hlubokého učení mimo brain-computer interface také aplikovaný na rozpoznávání obrazu i zvuku. Důležitou vlastností CNN je její schopnost rozpoznávat lokální příznaky. Díky této schopnosti je CNN invariantní vůči posuvu a je tedy vhodná pro použití na proměnlivém signálu jako EEG, který se může lišit člověk od člověka nebo i u stejného člověka den ode dne.

Konvoluční neuronová síť se skládá z několika speciálních vrstev. Kromě normalizační vrstvy a plně propojené neuronové vrstvy, které můžeme standardně najít u všech metod hlubokého učení, CNN obsahuje vrstvy konvoluční („convolution“) a agregační/sdružovací („pooling“). Často se používá větší množství konvolučních vrstev v řadě, které umožňují učení hlubších příznaků signálu.

Konvoluční vrstva provádí konvoluci nalezených příznaků a matice vstupů. Matice příznaků mají menší rozměry než matice vstupů. Konvoluce vstupu s příznaky generuje stejný počet matic výstupů jako je počet příznaků. Každý výstup obsahuje pravděpodobnosti výskytu příznaku na daném místě ve vstupu. Takto narůstá s každou konvoluční vrstvou počet zpracovávaných vstupů pro další vrstvu, agregační vrstvy ovšem tuto velikost redukuje.

Agregační vrstva redukuje rozměry vstupních matic tak, že s určitým krokem prochází podoblasti matice o určitých rozměrech a redukuje je na jeden výstup podle vybrané strategie. Běžnými strategiemi jsou minimální hodnota podoblasti, maximální hodnota podoblasti a průměrná hodnota podoblasti. Výhodou agregačních vrstev je krom redukce dat také jejich redukce přesnosti, díky této redukci je CNN schopná správně klasifikovat i drobné odlišnosti, které se takto odstraní.

## Support Vector Machine

Metoda podpůrných vektorů (SVM) je metoda strojového učení založená na lineární klasifikaci. Lineární předěl (nadrovina v nadprostoru) mezi třídami je určen největší vzdáleností od okrajových bodů (podpůrných vektorů) obou skupin, nebo je průměrován z více bodů, což umožňuje vyhnout se problémům vzniklým ze špatně klasifikovaných trénovacích dat. Důležitou součástí SVM je jádrová transformace („kernel transformation“) dat do vyšší dimenze. Takto lze lineárně klasifikovat dříve lineárně neoddělitelné třídy. Lze použít různé druhy jádrových transformací, nejčastěji polynomiální jádro, které vytváří vyšší dimenze

umocňováním vstupů, nebo radiální jádro, které pracuje s neomezeným počtem dimenzí a tvoří něco jako váhovací systém, kde třída nejbližších prvků má největší váhu.



## Kapitola 3

# Návrh a implementace aplikace

Cílem práce bylo vytvořit aplikaci spustitelnou na běžném počítači, která umožňuje psát na tomto počítači za použití mozkových signálů. Aplikace musí být schopná přijímat data z EEG a zpracovávat je na výstupy, také musí vhodnou formou komunikovat s uživatelem.

Pro implementaci aplikace byl zvolen programovací jazyk Python, který umožňuje vysokou úroveň abstrakce zejména na poli hlubokého učení. Python je jedním z jazyků podporovaných knihovnou *BrainFlow*<sup>1</sup>, která zajišťuje komunikaci se zařízením EEG a tvoří tedy základ projektu.

Tato kapitola popisuje návrh systému a jeho rozdělení do částí. Podrobně se věnuje grafickému rozhraní, které je důležité pro generování stimulů. Dále rozebírá detaily implementace na úrovni kódu, popisuje použité předzpracování signálu jednotlivých implementovaných systémů a nakonec rozebírá architektury použitých neuronových sítí a jiné použité klasifikační metody.

### 3.1 Návrh

Pro komunikaci s uživatelem je použito jednoduché grafické uživatelské prostředí skládající se z klávesnice generující potřebné vstupní mozkové signály a textového řádku, který informuje uživatele o napsaných písmenech a umožňuje tak uživateli reagovat na případné chyby. Grafické rozhraní běží paralelně s „backend“ částí a je jím tedy co nejméně ovlivňováno.

„Backend“ se skládá ze tří částí: *čtení signálu*, *předzpracování* a *klasifikace*. Ke čtení signálu musí docházet v reálném čase a signál musí být zařazen do správných datových struktur.

Předzpracování je nepovinnou součástí BCI a připravuje signál na jeho klasifikaci klasifikačním algoritmem v další části. Jedná se tedy zejména o filtrace nechtěných frekvencí, jako šum způsobený okolní elektrickou sítí nebo frekvencí považovaných za nepotřebné v rámci měření, které jsou výše identifikovány podle aktivity v podsece 2.1. V této části také dochází k transformaci signálu a extrakci příznaků signálu. Příznaky signálu jsou data reprezentující podstatné informace signálu, takto se redukuje množství dat pokračujících v systému do klasifikace.

Klasifikace je nejdůležitější a výpočetně nejnáročnější část programu. Vstupem jsou příznaky signálu a výstupem klasifikovaný znak z množiny možných znaků (z klávesnice). Klasifikace je implementována některou z klasifikačních metod strojového učení, musí tedy docházet ke kalibraci. Při použití hlubokého učení vstupem nebývají manuálně vybrané

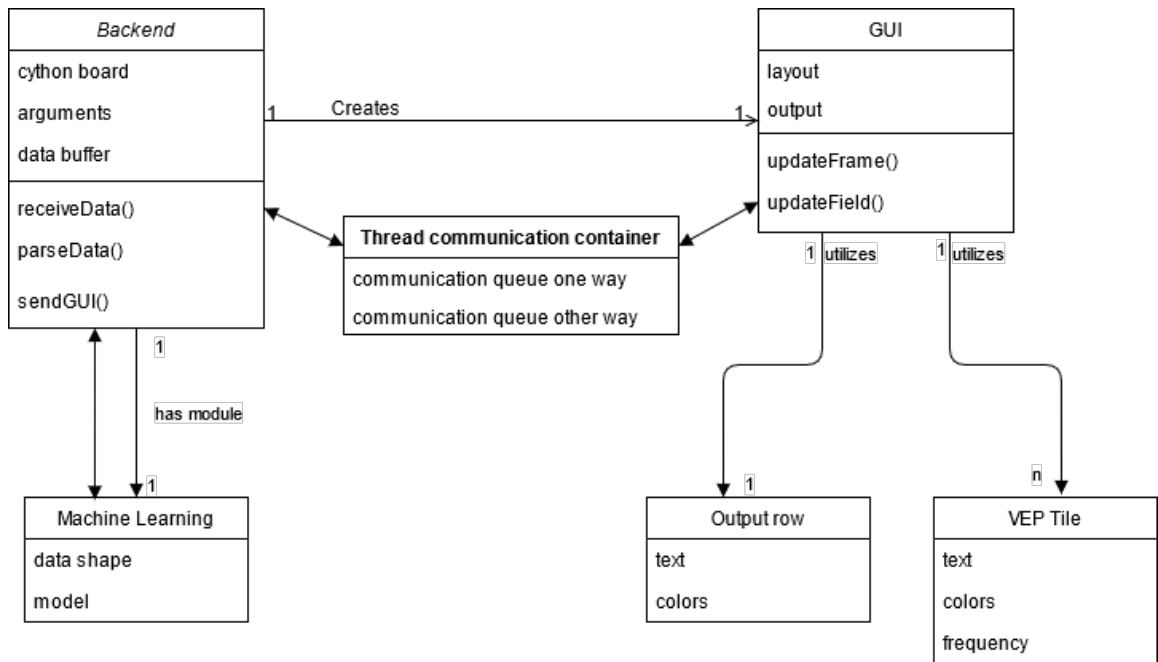
---

<sup>1</sup><https://brainflow.org>



příznaky, ale čistý, surový signál, k extrakci vhodných příznaků totiž dochází na skrytých vrstvách sítě.

Základní koncept rozložení systému lze vidět v diagramu tříd 3.1. Jelikož backend i grafické prostředí běží zároveň v reálném čase, je program rozdělen do dvou vláken. Komunikace mezi nimi je zajištěna jedinou třídou komunikačního modulu, který implementuje dvě fronty vstupů, každou pro jeden směr komunikace. Směr z backendu do grafického rozhraní zasílá požadavky o změnu rozhraní, jako aktualizace vypsaného řádku nebo krátká změna barvy při výběru znaku klávesnice. Směr opačný běžně signalizuje uživatelské vstupy, ale jelikož vstupy jsou snímány jiným zařízením, encefalografem, je tento směr vhodný pouze k signalizaci ukončení grafické části aplikace. Ve finální verzi byla třída informačního modulu odstraněna a nahrazena přímými odkazy mezi backend částí a grafickým rozhraním vždy na třídu druhou.



Obrázek 3.1: Diagram tříd znázorňující návrh systému a jeho komunikaci

Program má dva režimy, režim kalibrace a režim výkonný. V režimu kalibrace je úkolem uživatele pokoušet se opsat znaky z obrazovky, znaky jsou promítány na stejný řádek jako ve výkonném režimu vstupy, tyto znaky mění barvu a řádek se zkracuje za účelem signalizace postupu na znak další. Účelem kalibračního režimu je učení s učitelem pro neuronovou síť popřípadě jiné strojové učení. Tato fáze je kritická v problematice brain-computer interface, jelikož zpomaluje nasazení na uživateli, protože vyžaduje čas na kalibraci. Je proto častým předmětem optimalizačních strategií [22]. Ve výkonném režimu uživatel využívá aplikaci k libovolné komunikaci za použití klávesnice na obrazovce. Lze zapnout režim „text to speech“, ve kterém aplikace předčítá nahlas při potvrzení řádku.

## 3.2 Knihovny jazyka Python

Důležitou knihovnou, zmíněnou výše, je knihovna *BrainFlow*, která zajišťuje komunikaci mezi zařízením EEG OpenBCI a programem. Zařízení OpenBCI je napojeno na port USB,

odkud knihovna data čte. Knihovna umožňuje sběr signálu v reálném čase na všech kanálech současně. Knihovna dále poskytuje funkce pro některé operace nad daty, jako filtraci a transformace. Z těchto funkcí byla použita implementace filtru dolní propusti pro potlačení frekvencí vyšších 40 Hz.

Pro ostatní filtrování a transformování dat jsou použity knihovny *NumPy* a *SciPy*, které jsou standardem pro zpracovávání dat a signálů v jazyce Python. Všechna data jsou uložena ve struktuře pole knihovny NumPy, a tak je knihovna ideální volbou pro transformace dat jako transpozice za účelem ukládání dat do souboru nebo „baseline correction“ odečtením hodnot, tak aby se výsledný signál pohyboval kolem nuly. Modul *signal* knihovny SciPy umožňuje data transformovat do frekvenční domény použitím Fourierovy transformace. Modul je také použit k odstranění lineární složky signálu funkcí `detrend`. Z tohoto modulu byly také použity funkce realizující filtr pásmové propusti (`butter` a `sosfilt`).

Jelikož program realizuje jak grafické rozhraní tak vyhodnocení vstupů za použití neuronových sítí, program je rozdělen na současně běžící vlákna s použitím knihovny *Threading* a komunikaci mezi vlákny zajišťují fronty popřípadě atributy tříd, které jsou z vnějšku pouze pro čtení.

Hluboké učení implementuje knihovna *Keras* obsažená v balíčku *TensorFlow*, jedna z nejpoužívanějších knihoven jazyka Python pro strojové učení. Při trénování s uživatelem dochází k „online (inkrementálnímu) učení“ za účely okamžitého použití sítě ke klasifikaci výsledku, tak aby měl uživatel odezvu o průběhu trénování. Někteří uživatelé knihovny jsou skeptičtí k podpoře „online učení“<sup>2</sup>, protože údajně dochází k nevyžádané opakované inicializaci některých hodnot na hodnoty počáteční.

Je použita implementace metody podpurných vektorů knihovny *scikit-learn*, konkrétně `OneClassSVM`, která umožňuje trénování nad dávkou dat použitím funkce `fit()` a klasifikaci dat funkcí `predict`, důležitou použitou funkcí je `decision_function()`, která počítá vzdálenost klasifikovaných dat od trénovaného modelu.

Grafické rozhraní je vytvořeno v knihovně PyGame, původně určené k vývoji her. Pro tuto knihovnu jsem se rozhodl po krátkém průzkumu knihoven pro uživatelské prostředí. Ovšem žádná z takovýchto knihoven nepočítá s rapidně se měnícím rozhraním, jako je blikání bloků o zadané frekvenci. Jelikož herní knihovny implementují vykreslování i logiku na bázi snímků za sekundu, jsou vhodné pro zobrazení daných frekvencí.

Pro účely převádění textu na řeč je použita knihovna *pyttsx3*, která umožňuje převod provádět i bez připojení k internetu, ovšem proto knihovna potřebuje předinstalované některé balíčky operačního systému.

Také jsem uvažoval nad využitím knihovny modelů neuronových sítí<sup>3</sup> od *Army Research Laboratory*, nebo implementací jedné z testovaných sítí od Alvarado-González a kol.[2]<sup>4</sup>. Jejich implementace ovšem nakonec nebyly nepoužity.

### 3.3 Uživatelské rozhraní systému

Uživatelské rozhraní je pro c-VEP navrženo jako dlaždicová mříž o čtyřech řadách a šesti sloupcích s velkým písmenem uprostřed každé dlaždice. Počet řad a sloupců je odvozen od počtu požadovaných vstupů zvolené zobrazované „klávesnice“ a použité pseudonáhodné sekvence, která limituje počet rozpoznatelných vjemů. Délka takovéto sekvence určuje počet unikátních stavů a tudíž i počet unikátních stimulů a jejich vjemů. Jednotlivé dlaždice

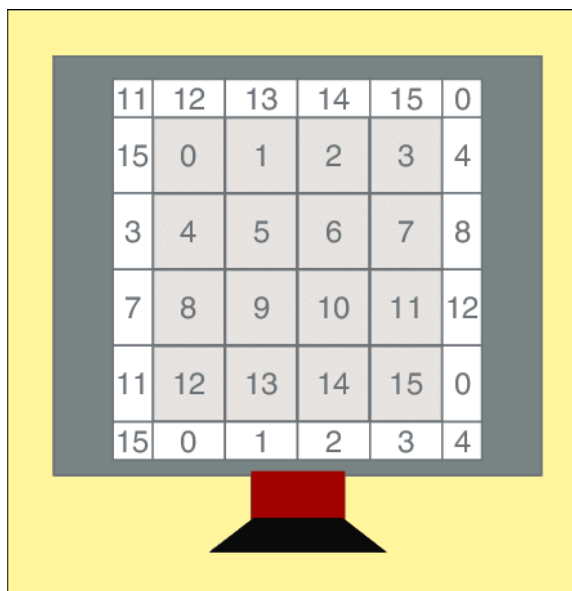
<sup>2</sup><https://github.com/keras-team/keras/issues/1868>

<sup>3</sup><https://github.com/vlawhern/arl-eegmodels>

<sup>4</sup><https://github.com/gibranfp/P300-CNNT>

invertují barvu se svým písmenem, takto je zajištěno blikání vyžadované pro tuto metodu. Další důležitou vlastností této mříže jsou blikající okraje. Okraje blikají identicky s písmenem na druhé straně obrazovky, jakoby byly dlaždice na rozvinutém toroidu (ovšem s jedním řádkem stočeným do spirály). Příklad rozložení dlaždic s okraji a znázorněnými identifikátory stejného posuvu blikání lze vidět na obrázku 3.2. Účelem těchto okrajů je zajištění stejných podmínek pro každý vjem. Pokud se subjekt soustředí na některý z plně obklopených stimulů, je lehce ovlivněn také blikáním stimulů sousedních, pokud by tedy okraje scházely, mohly by být krajní dlaždice špatně zařaditelné vzhledem k odlišnému šumu.

Metoda c-VEP byla implementována ve dvou „iteracích“, které se liší ve své komunikaci s grafickým rozhraním. První iterace se pokoušela co nejvíce oprostit od grafického rozhraní a chovala se tudíž asynchronně. Stimuly s daty synchronizovala dopočítáváním a dalším kruhovým posunem sekvence po klasifikaci. Druhá iterace eliminuje tento další matoucí posuv sekvence tím, že grafické rozhraní synchronizovala se sběrem dat. Během pauzy na změnu cíle tedy stimuly neblíkají a s každým začátkem sběru dat začínají vždy stejně.



Obrázek 3.2: Znázornění stejně blikajících okrajových polí. Dlaždice obsahují číslo identifikující unikátní blikání. Přičemž světlé okrajové dlaždice sdílí své blikání s blikáním dlaždice cílové. Převzato z [5]

Uživatelské rozhraní pro P300 není svým vzhledem odlišné od výše uvedeného rozhraní pro c-VEP, liší se ovšem svým chováním. V náhodném pořadí se rozsvěcují sloupce a poté řádky. Bylo zvažováno také použití úplně náhodného výběru, ale v takovém případě nemusí dojít k rozsvícení cíle, což by zbytečně snižovalo přesnost aplikace. Pokud ovšem zvolený přístup rozsvítí cílový stimul jako poslední z dané sekvence sloupců nebo řádků, tak již může uživatel plně očekávat výskyt cíle, což v takovém případě může snížit kvalitu signálu P300.

Uživatelské rozhraní metody SSVEP obsahuje pouze čtyři stimuly, které obsahují část abecedy. Při výběru stimulu dojde k dalšímu rozčtvrcení té části abecedy, kterou představuje, a v další iteraci stimuly představují právě tyto nové části. Pokud dojde k výběru části představující 1 písmeno, tak je písmeno vypsané a proces začne znovu s celou abecedou. Pro

jednoduchost implementace systém neumožňuje návrat o vrstvu výš, pokud dojde ke špatnému výběru. Taktéž je možné při postupném čtvrcení vytvořit stimuly nereprezentující žádný znak, protože délka použité abecedy není mocninou čtyř.

Součástí uživatelského rozhraní je také řádek reprezentující výstup, který zajišťuje komunikaci s uživatelem. Takto může uživatel odhalit špatně klasifikovaná písmena a pokusit se o jejich opravu. Tento řádek se také využívá při kalibrační fázi, a zobrazuje písmena, která se uživatel pokouší opsat, popřípadě jiné instrukce určující cíl soustředění.

## Symboly klávesnice

Symboly klávesnice se dělí na tisknutelné a kontrolní. Jak jejich název napovídá, tisknutelné se překládají přímo na výstup programu, zatímco kontrolními se aplikace ovládá.

Mezi tisknutelné symboly patří všechny znaky anglické abecedy (které jsou na dlaždicích označeny příslušným písmenem ve velké podobě) a mezera na dlaždici symbolizována podtržítkem.

Kontrolní symboly se skládají z mazání a potvrzení. Mazání slouží k odstranění posledního výstupu, většinou z důvodu špatné klasifikace. Potvrzení slouží k ukončení věty a vymazání celého řádku, pokud je aktivní „text to speech“, řádek je přečten nahlas. Na klávesnici je také zobrazen symbol nuly, který umožňuje do budoucna aplikaci rozšířit o přepínatelnou numerickou klávesnici.

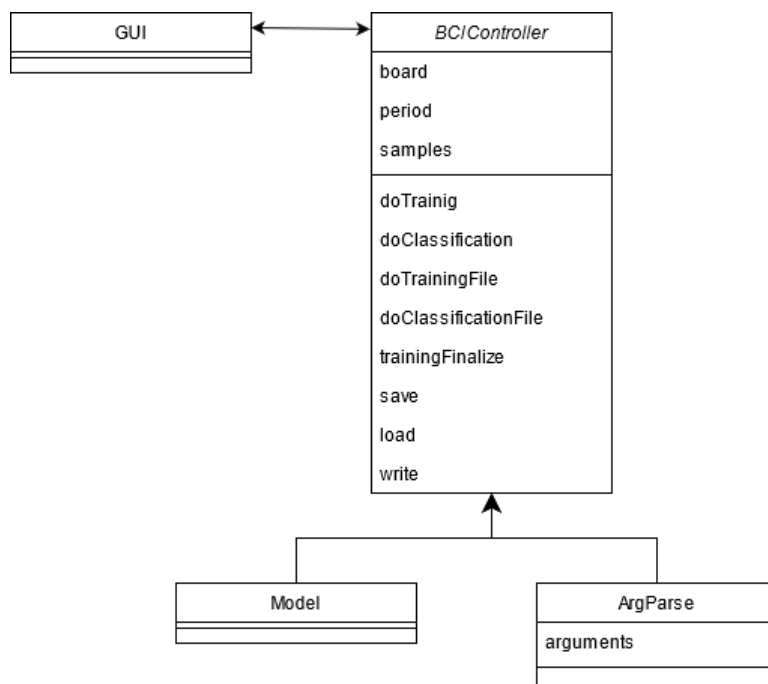
## 3.4 Detaily implementace

Mezi první a druhou iterací systému existuje několik rozdílů, které je zapotřebí zmínit. Hlavní změnou bylo odstranění třídy komunikačního modulu, který zajišťoval bezpečnou komunikaci mezi vlákny backend části a grafického rozhraní za použití front, které jsou v pythonu jednou z bezpečných metod komunikace mezi vlákny. V původním návrhu se komunikační modul skládal ze dvou front pro každý směr a každá strana musela v cyklu testovat, jestli na ni nečeká zpráva ve frontě. V systémech druhé iterace byl komunikační modul nahrazen řadou atributů z vnějšku jejich tříd určených pouze pro čtení a několika metodami. Konkrétně se jedná o atributy `run`, kterým grafické rozhraní signalizuje konec aplikace, `stimuli_run`, signalizující běh, či pauzu v blikání nebo pole `lighted` představující rozložení dlaždic a obsahující informaci o rozsvícení každé konkrétní dlaždice. Jednou ze zmíněných metod je metoda dlaždice `setSelect`, která umožňuje po zavolání řadičem („contoller“) nastavit dlaždici na krátkou, barevnou signalizaci uživateli, že byla vybrána jako klasifikovaný cíl.

Vstupní soubor programu (`main`) se stará o zpracování argumentů příkazové řádky, které určují vstupní soubory, výstupní soubor, načítání modelu ze souboru a úpravu některých konstant systému. Dále se také stará o inicializaci třídy `BoardShim` knihovny `Brainflow` a navázání spojení s EEG. Také inicializuje řadič a předává mu odkaz na objekt třídy `BoardShim`. Poté volá ve svém pořadí 4 funkce řadiče, představující fáze programu: `doFileTraining`, `doFileClassification`, `doTraining` a `doClassification`, které trénují a využívají model načtený řadičem. Fáze představující trénink a klasifikaci dat ze souboru mohou být systémem vynechány, pokud nejsou dané soubory specifikované v argumentech příkazové řádky.

Základní strukturu řadiče lze vidět na obrázku 3.3. Řadič si uchovává odkaz na objekt třídy `BoardShim`, přes který žádá o data z EEG, a který obsahuje základní informace o datech jako vzorkovací frekvenci a počet kanálů. řadič také obsahuje referenci na grafické

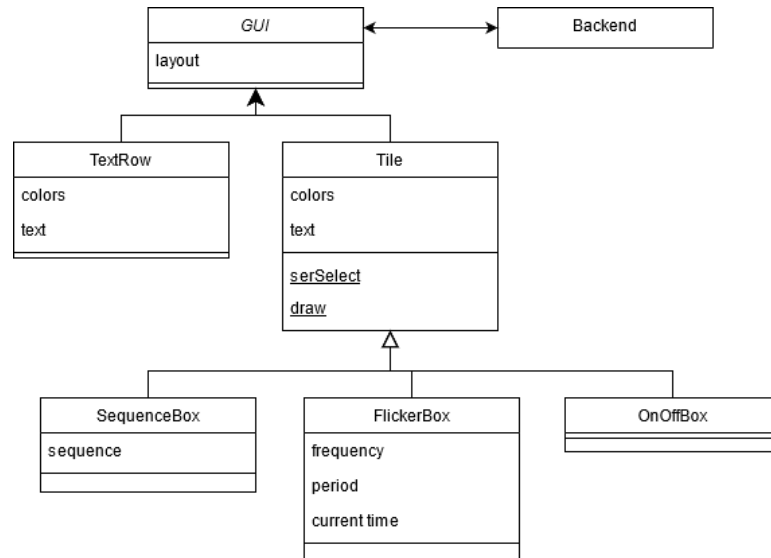
rozhraní, přes které ovládá zejména text dlaždic. K synchronizaci stimulů dochází za použití atributů, které jsou z pohledu grafického rozhraní pouze pro čtení. Tyto atributy si rozhraní kontroluje v každém cyklu samo. Mezi důležité metody patří metoda `__preprocess`, která je volána na všech místech, kde jsou načítána data buď z EEG, nebo i ze souboru a stará se o předzpracování signálu. Další důležité metody jsou metody `load` a `save`, které načítají model strojového učení, popřípadě inicializují proměnné jiných klasifikačních metod a umožňují model uložit do souboru. Vzhledem ke specifickým požadavkům jednotlivých paradigmat stimulů má každý systém vlastní třídu řadiče.



Obrázek 3.3: Znázornění důležitých struktur a metod řadiče

Implementační struktura grafického rozhraní je znázorněna na obrázku 3.4. Třída dědí ze třídy `Thread` a umožňuje tedy svému objektu spuštění na odděleném vlákne, přičemž si zachovává svůj status objektu a může tedy přechovávat atributy, které lze modifikovat za běhu z vnějšku. V původním návrhu se jednalo o funkci, která tedy potřebovala pro komunikaci výše zmíněný komunikační modul. Změna na třídu umožnila odstranění tohoto modulu a zjednodušení komunikace. Metoda `run` obsahuje hlavní smyčku grafického rozhraní, která periodicky vykresluje dlaždice na obrazovku s cílovou obnovovací frekvencí 60 Hz. Tato metoda obsahuje konstanty určující rozložení dlaždic při vykreslení, tyto konstanty mohou být pro jednotlivé systémy (zejména SSVEP) odlišné. Jediným významným atributem třídy je pole obsahující zobrazované dlaždice, které přímo upravuje řadič.

Implementace každého paradigmatu stimulu je obsaženo ve své vlastní třídě představující dlaždici grafického rozhraní. Tyto třídy implementují metodu `draw`, která vykresluje dlaždici na daných souřadnicích okna, za použití vykreslení obdélníku a vykreslení textu z knihovny `pygame`. Tato metoda přijímá mezi parametry změnu času, na které je založen tzv. `game loop` knihovny `pygame`. Tento parametr umožňuje dlaždicím „blikat“ nezávisle na řadiči a udržet si stabilní frekvenci. Za účelem udržení obnovovací frekvence `game loop` jsou výpočty v cyklu silně optimalizovány tak, aby obsahovaly co nejméně podmíněných větví programu a složitých operací.



Obrázek 3.4: Znázornění důležitých struktur a metod grafického rozhraní

Dlaždice pro *c-VEP* při své inicializaci ukládají *m*-sekvenci již v posunutém stavu. Jelikož délka sekvence a její perioda je konstantní, a tedy i čas na jeden bit sekvence je konstantní, pro všechny dlaždice stačí v jednom průběhu cyklu pro všechny dlaždice vypočítat index sekvence v daném časovém posunutí v periodě pouze jednou. Během vykreslování lze pak jednoduše použít stejný index u všech dlaždic, protože jak bylo zmíněno výše dlaždice si ukládají sekvenci již posunutou.

Dlaždice pro *SSVEP* si nápodobně ukládají při své inicializaci frekvenci, ze které lze spočítat periodu a půl-periodu. Půl-perioda v periodě značí světelný stav dlaždice, v první půl-periodě dlaždice svítí a v druhé je dlaždice tmavá, takto je vytvořeno blikání o konstantní frekvenci. Při vykreslování pak stačí určit ve které polovině periody se současný stav periody nachází.

Dlaždice implementující *P300* naopak potřebují signál z řadiče, takže samy o sobě se chovají nejjednodušeji a signál k rozsvícení přijímají v parametrech vykreslovací metody. Řadič tento signál ukládá pro jednotlivé dlaždice do pole ve formě booleovské hodnoty a při vykreslování je dlaždicím přiřazen signál odpovídající jejich pozici v grafickém rozhraní.

### 3.5 Předzpracování a generování dat

Sběr dat probíhá za použití zařízení OpenBCI, výše zmíněného v sekci 2.1, se vzorkovací frekvencí 125 Hz na šestnácti kanálech.

Signál je před klasifikací předzpracován za účely potlačení šumu. První iterace systému nejprve provádí jednoduchou „*baseline correction*“ a od každého vzorku je odečtena hodnota prvního vzorku, takto je zajištěno, že signál začíná na nule a může být bez problému proveden „*lowpass filter*“. „*Lowpass filter*“ je typu *Butterworth* pátého řádu a hranici má nastavenou na 40 Hz. Nakonec je pro účel korelace ze signálu odstraněn lineární trend.

U systémů druhé iterace je proces filtrace následující: Namísto jednoduchého „*baseline correction*“ dochází k odstranění lineárního trendu. V případě systému *SSVEP* je použito *Hammingovo okno*. Následně je aplikován filtr pásmové propusti taktéž typu *Butterworth* pátého řádu, s frekvencemi od 1 Hz do 20 Hz pro systém *P300*, od 0,1 Hz do 30 Hz pro systém

c-VEP a od 4 Hz do 30 Hz pro systém SSVEP. K těmto hodnotám se došlo experimentálně testováním s ideálním signálem. Pro systém SSVEP také dochází k převodu signálu do frekvenčního spektra rychlou Fourierovou transformací. Nakonec je signál normalizován vydělením každého vzorku svým vzorkem s nejvyšší hodnotou.

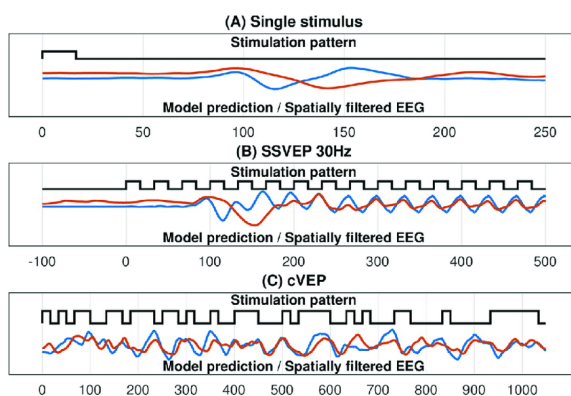
Metoda kódově modulovaného vizuálně evokovaného potenciálu obsahuje na jednu epochu sběru dat 3 sekundy signálu, ale bylo experimentováno i se 2 sekundami a 2.5 sekundami. Tato délka odpovídá jedné periodě sekvence stimulu, po sběru následuje sekundová pauza, která umožňuje subjektu změnit soustředění na další cíl. Sekvence v první iteraci obsahuje 31 bitů a klávesnice 30 cílů, zatímco v druhé iteraci za účely snazší rozlišitelnosti je použita sekvence o 63 bitech a její kruhový posuv po 2 bitech nebo 4 bitech, tato sekvence je dvakrát delší a tím pádem se zpomaluje celkový přenos informace.

Pro metodu P300 je délka jedné epochy půl sekundy, ale bylo experimentováno s délkou celé sekundy. První čtvrtinu tohoto času je některá skupina stimulů rozsvícena a po zbytek času jsou všechny stimuly tmavé.

Metoda SSVEP sbírá data z jednoho cíle po dobu 4 sekund, bylo experimentováno s použitím celé periody, ale i plovoucího okna o délce 2 sekundy a posuvem 50 vzorků, takto je signál efektivně rozdělen na několik pod-signálů a navýšen počet trénovacích vzorků.

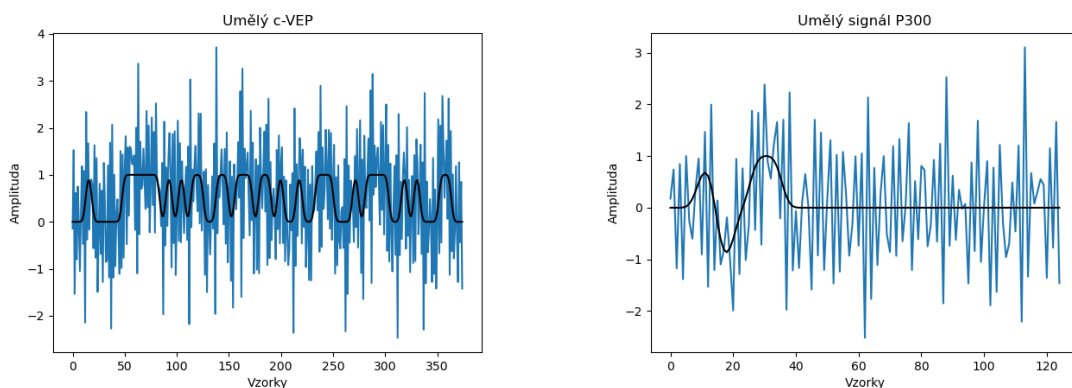
## Umělé signály

Pro účely testování byly implementovány generátory syntetických signálů. Tyto signály by měly představovat přibližně ideální vstupy daných systémů. Základem umělých signálů jsou očekávané evokované odezvy při použití daných stimulů. V praxi to znamená, že systém SSVEP využívá signál založený na dominantní frekvenci, systém P300 signál založený na P2, N3 a P3 vlnách odezvy, které nastávají zhruba ve 100 ms, 150 ms a 250 ms v uvedeném pořadí. a systém c-VEP zakládá signál na prosté odezvě ve formě jedné vlny na pozitivní změnu v sekvenci. Tyto odezvy je možné vidět na obrázku 3.5. Do umělého signálu je zanesen šum použitím náhodných odchylek vzorků za pomoci normálního rozložení. Dále je přidán signál o 50 Hz představující rušení způsobené elektrickou sítí a signál 60 Hz a jeho slabší subharmonický 30 Hz představující obnovovací frekvenci monitoru. Při syntetizaci signálu SSVEP jsou frekvence vedlejších cílů přítomné ve snížené amplitudě jako šum signálu. Příklady umělých signálů jsou zobrazeny na obrázku 3.6



Obrázek 3.5: Odezvy evokované v signále použitím různých paradigmatů, černě stimulus, červeně EEG, převzato z [15]





Obrázek 3.6: Syntetizované signály c-VEP a P300 se základní odezvou (černě)

### 3.6 Vlastní dataset

Program umožňuje ukládat epochy dat do souboru ve formátu *csv* s lehkými úpravami. Pro účely komentování lze v souboru začít řádek symbolem křížku (#), tyto řádky budou při čtení souboru ignorovány. Soubor také obsahuje řádky začínající symbolem zavináče (@), tyto řádky obsahují informaci o cílovém stimulu na který se subjekt soustředil. V případě P300 tento řádek lze spolehlivě nastavit pouze během trénování a obsahuje informaci booleanové hodnoty o tom, že data by měla obsahovat P300, mimo trénování je řádek prázdný. V případě kódově modulovaného potenciálu tento řádek v první iteraci systému informuje o posunutí sekvence grafického rozhraní vůči začátku sběru dat a v obou iteracích systému cílové písmeno, které je zapotřebí při zápisu vyplnit ručně. Řádek informující o cíli je následován daty, které popisuje. Každý řádek dat reprezentuje jeden vzorek všech šestnácti kanálů, to znamená, že data jsou v souboru ukládána transponovaná oproti vnitřní reprezentaci dat používané knihovny. Takto je zaručena přehlednost a jednodušší možnost úprav.

### 3.7 Využití strojové učení

V oblasti BCI se často používají konvoluční neuronové sítě. Před vstupem do sítě se data transformují tak aby byly příznaky signálu co nejlépe čitelné. V případě metody SSVEP tedy do domény frekvenční, nebo jako v případě metody P300 se ponechávají v doméně časové. Pokud je použita metoda kombinací různých frekvencí a různých fází stejných frekvencí, lze použít i dvourozměrnou konvoluci v doméně časově-frekvenční a převést data do spektrogramu.

Konvoluční neuronové sítě se využívají jako trénovatelné filtry nad daty. Zejména filtr *časový* jako konvoluce přes signál v čase, jehož úkolem je nalézt vzor v signálu, a filtr *prostorový* přes všechny elektrody naráz, který váhuje vstupy z jednotlivých elektrod. Některé často používané sítě používají první konvoluční vrstvu právě jako prostorový filtr, ale takto je možné, že se ztrácí vzor signálu z jednotlivých elektrod. Tak vznikly sítě, které začínají filtrováním časovým[19][20].



## CNN pro c-VEP

Pro metodu c-VEP systém testoval hned několik modelů neuronových sítí, protože použití neuronových sítí pro metodu není běžné. Jejich struktury jsou znázorněny v tabulkách 3.1, 3.2 a 3.3. Architektury jsou odvozeny od návrhu Sebastian Nagel a kol.[16] a Hongchang Shan a kol.[19], tyto sítě sice detekují pouze existenci jedné hrany, ale představují základ zpracování vizuálně evokovaných potenciálů EGG.

Cílem neuronové sítě je klasifikace signálu do 31 tříd reprezentujících posunutí sekvence stimulu. Představa účelů vrstev spočívá v aplikaci prostorového filtru a filtru časového obsahujícího 1 bit sekvence, který reprezentuje světlý nebo tmavý stav, na dalších hlubších vrstvách by síť určovala pořadí světlých a tmavých bitů, tak aby odpovídaly trénované sekvenci v daném posunutí. Teorie byla otestována na samotné m-sekvenci rozšířené na větší počet vzorků s přidaným náhodným šumem a posunutím. Při použití na samotné m-sekvenci neměly sítě problém naučit se posloupnost a posunutí sekvence a dosahovaly přesnosti přes 80%. Žádná ze sítí ovšem nedosáhla očekávaných výsledků na reálných datech. Důvodem může být příliš velký poměr šumu a signálu, nevhodná konstrukce stimulů nebo také špatný návrh nebo trénování na malém množství dat.

V tabulce 3.1 lze vidět strukturu neuronové sítě se dvěma konvolučními vrstvami s jádrem ve směru časové osy. Cílem bylo naučit neuronovou síť rozpoznávat větší části signálu a podle nich určovat posunutí sekvence. Problém tohoto přístupu spočívá v okrajích sekvence, které předělují subsekvence při každém posuvu a znemožňují tak jejich rozpoznání. Pokud se ale síť naučí rozpoznávat alespoň pár nepřekrývajících se klíčových subsekvencí v sekvenci, může klasifikovat jejich posunutí v rámci signálu, i když bude jedna z nich předělena kruhovým posuvem.

| Operace            | Jádro/Parametry    | Aktivace |
|--------------------|--------------------|----------|
| Input              | (C, S, 1)          |          |
| Conv2D             | $16 \times (C, 1)$ | ReLU     |
| Reshape            | (S, 16)            |          |
| MaxPooling1D       | (2)                |          |
| BatchNormalization |                    |          |
| Conv1D             | $16 \times (S/31)$ | ReLU     |
| MaxPooling1D       | (2)                |          |
| Conv1D             | $32 \times (5)$    | ReLU     |
| MaxPooling1D       | (2)                |          |
| Flatten            |                    |          |
| Dense              | 31                 | Softmax  |

Tabulka 3.1: Síť s využitím dvou konvolučních vrstev

V tabulce 3.2 je znázorněna neuronová síť se skrytou plně propojenou vrstvou následující po konvoluci v časové ose. Účelem je ponechání dostatku uzlů na určení složitých struktur reprezentujících stavu signálu. Vzhledem k použití aktivační funkce „rectified linear unit“ by mělo časem dojít k odumření bezvýznamných uzlů.

Pro dané sítě bylo experimentováno s parametry, jako například velikost konvolučního jádra, počtem filtrů nebo počtem uzlů. Taktéž byly otestovány přímo navržené sítě od Sebastian Nagel a kol.[16] a Hongchang Shan a kol.[19] bez vnitřních změn, jedinou změnou byla změna počtu výstupních tříd. Jak bylo zmíněno výše, žádná z otestovaných sítí nedosáhla na datech rozumných výsledků.

| Operace            | Jádro/Parametry    | Aktivace |
|--------------------|--------------------|----------|
| Input              | (C, S, 1)          |          |
| Conv2D             | $16 \times (C, 1)$ | ReLU     |
| Reshape            | (S, 16)            |          |
| MaxPooling1D       | (2)                |          |
| BatchNormalization |                    |          |
| Conv1D             | $16 \times (S/31)$ | ReLU     |
| MaxPooling1D       | (2)                |          |
| Flatten            |                    |          |
| Dense              | 1024               | ReLU     |
| Dense              | 31                 | Softmax  |

Tabulka 3.2: Síť s plně propojenou skrytou vrstvou

| Operace | Jádro/Parametry       | Aktivace |
|---------|-----------------------|----------|
| Input   | (C, S, 1)             |          |
| Conv2D  | $16 \times (C, S/31)$ | ReLU     |
| Dropout | 0.25                  |          |
| Flatten |                       |          |
| Dense   | 31                    | Softmax  |

Tabulka 3.3: Síť o jedné vrstvě

V prvních experimentech s neuronovou sítí v metodě c-VEP síť přijímala na vstupu krom dat také posunutí sekvence grafického rozhraní vůči začátku snímání dat. Tato informace byla připojena k výstupům ze skrytých vrstev jako prosté číslo, kterého měla využívat vrstva finální představující klasifikaci pro posun. Jelikož se sítě nebyly schopny nic naučit, tak zpracování této informace bylo za účelem zjednodušení přemístěno do klasifikace cíle výstupu sítě. Sítě se ovšem nebyly schopny nic naučit ani po manuálním přepočítáváním posunutí. Jelikož doposud se grafické uživatelské rozhraní chovalo částečně asynchronně, bylo možné, že data byla náhodně posunuta v rámci jednoho bitu sekvence, čímž se zanášel zbytečný šum a v další iteraci programu bylo grafické rozhraní předěláno na plně synchronní. Ani takto ovšem žádná ze sítí negenerovala smysluplné výsledky.

## CNN pro P300

Metoda P300 využívá modelu založeném na „Jednoduchém jedno vrstevném modelu“ navrženém Hongchang Shan a kol.[19]. Model se relativně rychle trénuje, díky svému relativně malému množství trénovatelných parametrů a dosahuje přesnosti 80%[2]. Existují modely rychlejší i přesnější, jako třeba „DeepConvNet“ s 84% přesností[2], ale žádný jiný model neposkytuje takto jednoduchou a průhlednou architekturu.

Hlavní rozdíl od modelu navrženého Hongchang Shan a kol.[19] spočívá v použití chybové funkce „Categorical cross entropy“, která je doporučena pro více než jeden neuron ve výstupní vrstvě, přestože je „binary cross entropy“ speciálním případem funkce výše zmíněné, dokumentace knihovny keras se zmiňuje o rozdílu v očekávaném kódování „labelů“, první výše zmíněná očekává „One-Hot“ kódování, zatímco druhá jedno číslo s plovoucí řádovou čárkou. Další rozdíl tvoří použití optimalizační metody „stochastic gradient descent“ bez setrvačnosti (momentum) a velikost dávky trénovacích dat 1, za účely trénování na živých datech.

| Operace | Jádro/Parametry       | Aktivace |
|---------|-----------------------|----------|
| Input   | (C, S, 1)             |          |
| Conv2D  | $16 \times (C, S/15)$ | ReLU     |
| Dropout | 0.25                  |          |
| Flatten |                       |          |
| Dense   | 2                     | Softmax  |

Tabulka 3.4:

Také byla otestována síť navržená Gunsteren, na kterou jsem narazil v [16], jejíž architekturu lze vidět v tabulce 3.5.

| Operace            | Jádro/Parametry    | Aktivace |
|--------------------|--------------------|----------|
| Input              | (C, S, 1)          |          |
| Conv2D             | $16 \times (C, 1)$ | ReLU     |
| Reshape            | (S, 16)            |          |
| MaxPooling1D       | (2)                |          |
| BatchNormalization |                    |          |
| Conv1D             | $16 \times (16)$   | ReLU     |
| MaxPooling1D       | (2)                |          |
| BatchNormalization |                    |          |
| Conv1D             | $8 \times (5)$     | ReLU     |
| MaxPooling1D       | (2)                |          |
| Flatten            |                    |          |
| Dense              | 1024               | ReLU     |
| Dropout            | 0.5                |          |
| Dense              | 2                  | Softmax  |

Tabulka 3.5: Síť založená na síti navržené Gunsteren 2018

## CNN pro SSVEP

Metoda využívá síť „CNN-1“ navržené No-Sang Kwak a kol.[11], která překonává ostatní navržené metody strojového učení a dosahuje přesnosti přes 99%. Její architektura je znázorněna v tabulce 3.6. Využívá dvou konvolučních vrstev, první z nich funguje jako filtr prostorový a druhá jako filtr časový. Síť neobsahuje žádné další, hlubší vrstvy.

## Jiné metody klasifikace

Pro metodu kódově modulovaného vizuálně evokovaného potenciálu, byly otestovány také jiné způsoby klasifikace, za účely srovnání výsledků s neuronovou sítí. Konkrétně výše zmíněná naivní korelace se samotnou sekvencí a korelace se vzorkem získaným průměrováním. Mezi časté metody klasifikace c-VEP také patří „one class support vector machine“, která byla otestována pouze „off-line“ na získaných datech.

Při naivní klasifikaci byl signál čisté sekvence generován rozšířením základní sekvence o 31 bitech na počet vzorků signálu zachyceného. Ovšem počet vzorků signálu nemusí být násobkem počtu bitů, pro tento případ je generace sekvence podobná generaci sekvencí v grafickém rozhraní a s využitím vzorkovací frekvence hardware jsou bity rozšířeny na jednotlivé vzorky. Takto je vytvořena sekvence o stejné délce jako data. Rozšířená sekvence

| Operace            | Jádro/Parametry    | Aktivace |
|--------------------|--------------------|----------|
| Input              | (C, S, 1)          |          |
| Conv2D             | $16 \times (C, 1)$ | ReLU     |
| Reshape            | (S, 16)            |          |
| MaxPooling1D       | (2)                |          |
| BatchNormalization |                    |          |
| Conv1D             | $8 \times (11)$    | ReLU     |
| MaxPooling1D       | (2)                |          |
| Flatten            |                    |          |
| Dense              | 4                  | Softmax  |

Tabulka 3.6:

je vygenerována pro všech 31 posunutí a následně je vypočítána jejich vzájemná korelace s daty funkcí *correlate* knihovny *SciPy*. Z těchto 31 korelačních koeficientů představujících posunutí prodloužené sekvence je vybrán ten s nejvyšší hodnotou. Od tohoto posunutí je následně odečteno posunutí grafického rozhraní vůči začátku sběru dat, jelikož byla naivní klasifikace použita pouze s asynchronním grafickým rozhraním.

Podobným způsobem se klasifikuje průměrovaným vzorkem. Při trénování dochází k váhovanému průměrování uloženého vzorku a současných dat, kde za každou iteraci trénování má vzorek o jedna vyšší váhu než data a začíná na nule. Takto mají všechna data stejnou váhu jako data předchozí. Průměrováním dochází k eliminaci šumu a zvýraznění segmentů podobných mezi signály. Při ukončení tréninku dojde k vytvoření dalších třiceti vzorů posunutých na jednotlivé bity sekvence, popřípadě je vytvořeno vzorů méně podle počtu cílů v klávesnici a jejich posunutí, jelikož často nejsou použity všechny bity sekvence. Ke klasifikaci dojde stejným způsobem, výběrem nejvyššího korelačního koeficientu mezi některým ze vzorů a příchozími daty.

Použití „one class support vector machine“, jak bylo zmíněno výše, bylo otestováno „off-line“ na uložených datech z druhé iterace systému c-VEP. Nevýhodou OCSVM je nutnost trénování na úplném datasetu a neumožňuje inkrementální trénink. Z datasetu bylo tedy použito 90 vzorků dat na trénink a 30 vzorků na validaci. Pro SVM bylo použito lineární jádro, které dosahuje nejlepších výsledků (jak bylo ověřeno v [3]). OCSVM má podobnou funkci jako korelace se vzorkem a počítá vzdálenost od naučeného vzoru. Ke klasifikaci tedy dojde určením nejmenší vzdálenosti od jednotlivých posunutých vzorků. Pro jednoduchost implementace je využito kruhového charakteru sekvence a dochází k posuvu signálu a jeho testování vůči jednomu naučenému modelu.

## Shrnutí

Pro všechny metody byly otestovány možnosti použití neuronových sítí. Žádná ze sítí ovšem nedosáhla očekávaných výsledků. Je možné, že toto selhání způsobil pokus o inkrementální/on-line učení neuronových sítí. Takto jsou data trénována na dávce o velikosti jedna a síť je trénována při každém novém vstupu. Při takovém trénování nelze využívat optimalizační strategie a je možné, že starší trénovaná data ztrácí svou váhu<sup>5</sup>. Ovšem tato obava byla vyvrácena při „off-line“ testování, které umožnilo upravit učení neuronových sítí k použití dávek a dynamických optimizátorů jako *Adam* a *Adagrad*.

<sup>5</sup><https://github.com/keras-team/keras/issues/1868>

Ani klasifikační metoda využívající průměrování nedosáhla žádného výsledku. Je možné, že toto selhání způsobil malý počet vzorků, ale selhání všech metod naznačuje, že by chyba mohla být někde v částech společných mezi metodami.

## Kapitola 4

# Testování systému na subjektech

Vzhledem k pandemické situaci byl systém testován na jediném subjektu. Jednalo se o zdravého muže ve věku 22 let. Subjekt pozoroval obrazovku ze vzdálenosti 30 cm. Celkem proběhla čtyři sezení, během kterých byl systém testován.

Na sezení při kterém byl systém testován prvně došlo ke kolekci dat, jelikož tato data jsou ukládána za běhu programu, je první test vždy jednoduchým testem funkčnosti a jeho výsledky lze odečíst z výstupů systému. Většina testů poté může být provedena nad uloženými daty „off-line“. Tato data byla dále analyzována na grafech v časovém nebo frekvenčním spektru, podle očekávaných příznaků v jednom nebo druhém. Pro systémy c-VEP a P300 byly krom jednotlivých vzorků analyzovány také jejich průměry, které by měly v těchto systémech redukovat šum signálu a vytvořit jeden ideální vzor signálu. S použitím těchto analýz bylo pro každý systém experimentováno s různými postupy předzpracování signálu.

K analýze jednotlivých klasifikačních algoritmů došlo použitím části uložených dat jako dat trénovacích a jiné části jako dat verifikačních. Trénovací data byla náhodně skládaná podle cílů, aby se zabránilo trénování na jediný výsledek klasifikace. Ve verifikačních datech bylo dbáno na poměry cílových tříd, protože tyto poměry se mohou promítnout do celkové přesnosti klasifikace při klasifikaci plně náhodné.

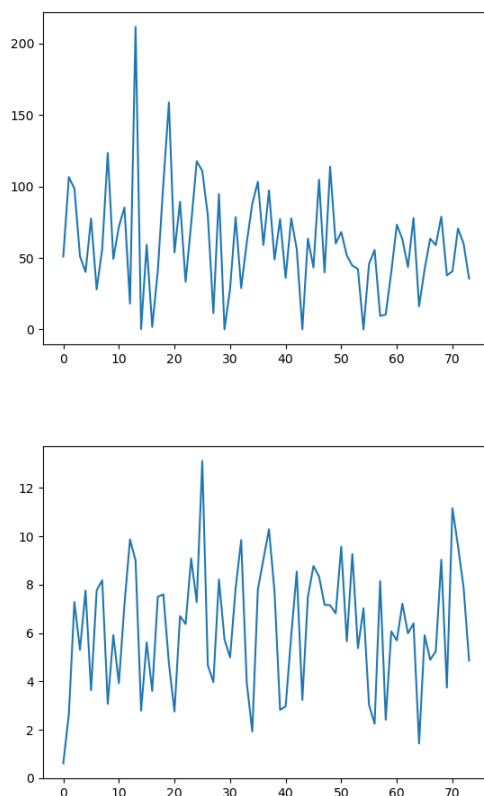
Pro neuronové sítě byly promítány do grafů jejich přesnosti a chybové funkce, ze kterých lze vyčíst, jestli daná síť konverguje k řešení. U klasifikace využívající SVM nebo korelaci s průměrem byly sledovány jejich přesnosti na celé dávce a rozdíly ve výsledných vzdálenostech mezi cíli v jedné epoše dávky.

Pro každý systém byl taktéž implementován generátor syntetického signálu, který by měl představovat přibližně ideální vstup systému i se simulovaným šumem. Tento signál byl užitečný při testování předzpracování, protože z jeho analýzy lze spolehlivěji vyčíst přítomnost příznaků po filtraci. Další využití má tento signál při validaci konceptů klasifikačních algoritmů, jako signál ideální nemůže zcela potvrdit použitelnost klasifikačních algoritmů, ale lze jej využít právě pro potvrzení funkčnosti základní myšlenky daného algoritmu.

### 4.1 Výsledky první iterace systému c-VEP

První iterace systému c-VEP byla testována ve dvou sezeních, při každém sezení byla použita data uložena ve formátu výše zmíněném v sekci 3.6. Při prvním sezení byly provedeny pouze tři pokusy bez chyby v soustředění, které byly uloženy do jednotlivých souborů pro účely dalšího „off-line“ testování. Při druhém byly podobně vytvořeny 4 další záchyty dat.

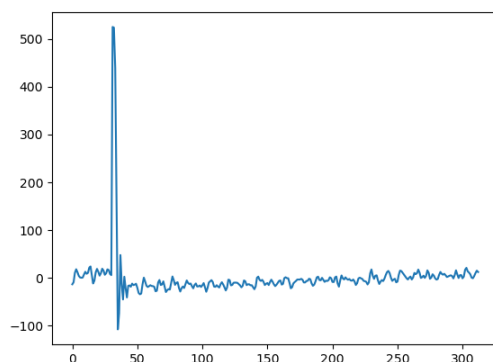
Během prvního sezení bylo otestováno použití neuronové sítě vlastního návrhu, která nepoužívala prostorový filtr. Průběh jejího učení je znázorněn v horním grafu obrázku 4.1. Z průběhu grafu bylo usouzeno, že síť nekonvergovala, a nemohla tedy dosáhnout požadovaných výsledků, pokud by se síť blížila k určitému výsledku, chybová funkce by konvergovala k 0. Z tohoto důvodu byly při dalších sezeních i „off-line“ otestovány sítě jiné. Chybová funkce jedné z nich je znázorněna na spodním grafu téhož obrázku 4.1. Žádná z otestovaných sítí s časem nekonvergovala, a nebyly tedy použitelné pro účely klasifikace.



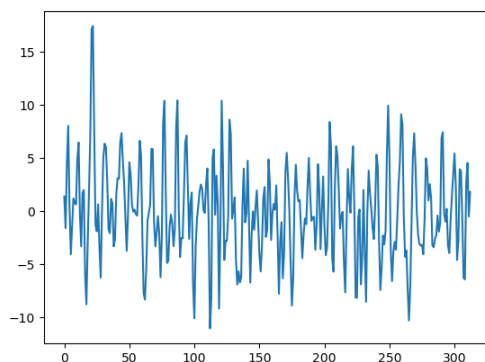
Obrázek 4.1: Chybové funkce některých použitých neuronových sítí

Při „off-line“ testování se odstoupilo od neuronových sítí a začalo se používat běžné metody průměrování signálu za účelem extrakce „čistého“ vzoru signálu obsahujícího sekvenci pro klasifikaci korelací. Při použití této metody byla objevena chyba vznikající pomalou inicializací čipu EEG, která způsobovala pomalý nárůst naměřených hodnot od nuly do hodnot skutečných a takto znehodnocovala první sběr dat uskutečněný ihned po inicializaci. Pro neuronovou síť by neměl být tento problém velmi relevantní, i přesto byly po opravě sítě otestovány znovu, opět bez úspěchu. Tento problém se ovšem znatelně projeví při průměrování signálu a vytvoří špičku ve vzoru, jak lze vidět na obrázku 4.2.

Vzor získaný průměrováním šedesáti epoch lze vidět na obrázku 4.3. Epochy by měly obsahovat odezvu na m-sekvenci o délce 31 bitů, běžně by měla odezva přibližně kopírovat sekvenci asi se 100 ms zpožděním, jak je zmíněno v [16], což odpovídá asi 13 vzorkům. Signál je ovšem příliš zašumělý a nelze spolehlivě určit špičky, které by představovaly změny v sekvenci. Je možné, že by šum dále klesal při dalším průměrování. Výsledky generované korelací tohoto vzoru taktéž nedosáhly výsledků, které lze označit za lepší než náhodné.



Obrázek 4.2: Chyba ve vzoru způsobená inicializací EEG



Obrázek 4.3: Vzorový signál vyprodukovaný z 30 cílů

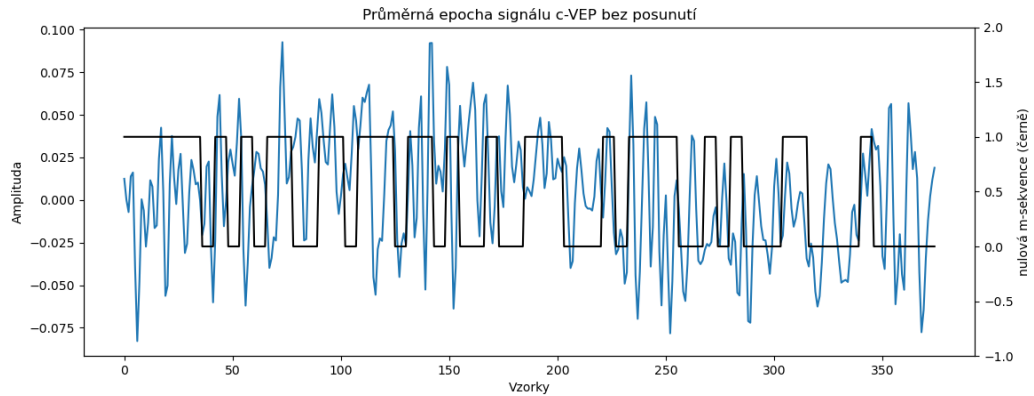
## 4.2 Výsledky druhé iterace systému c-VEP

Druhá iterace, jak bylo popsáno výše, se odlišuje od první použitím synchronního grafického rozhraní, které by mělo generovat signál s menšími odchylkami než při použití asynchronního rozhraní. Jelikož systém nemusí ukládat časové posunutí grafického rozhraní, liší se i formát uloženého souboru. K uložení nového „datasetu“ došlo v rámci jednoho sezení a obsahuje přibližně 120 sekvencí rozdělených do tří souborů.

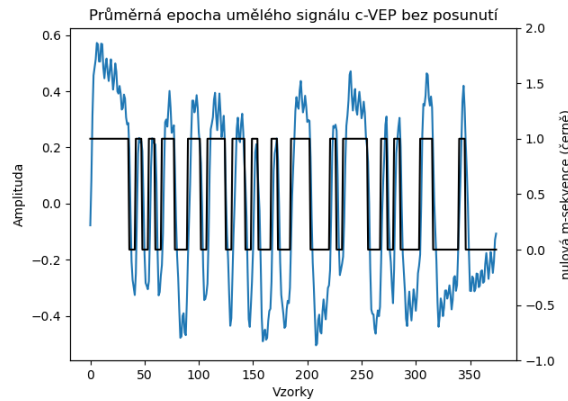
Výsledný vzor vyprodukovaný z těchto 120 cílů lze vidět na obrázku 4.4 spolu s m-sekvencí, kterou představuje. Ze signálu vzoru vystupují značné špičky, ovšem nezdá se, že by odpovídaly hranám sekvence, dále je možné vypořádat, že žádná ze špiček se nenachází na některém z nadějných míst ve vzoru vyprodukovaném předchozí verzí systému z obrázku 4.3.

Ani tato verze systému s tímto vzorem nedosáhla výsledků lepších než náhodných na signálech skutečných. Ovšem při práci se signálem umělým, jehož struktura byla popsána výše v sekci 3.5, byl systém schopen dosáhnout 100% přesnosti klasifikace a vytvořil okem rozpoznatelný vzor, zobrazený v grafu 4.5. Skutečný signál ovšem vytváří lehce odlišné odezvy v signále, které jsou vůči svému stimulu posunuty v čase zhruba o 100 ms.





Obrázek 4.4: Vzorový signál vyprodukovaný ze 120 cílů, zobrazený spolu s m-sekvencí

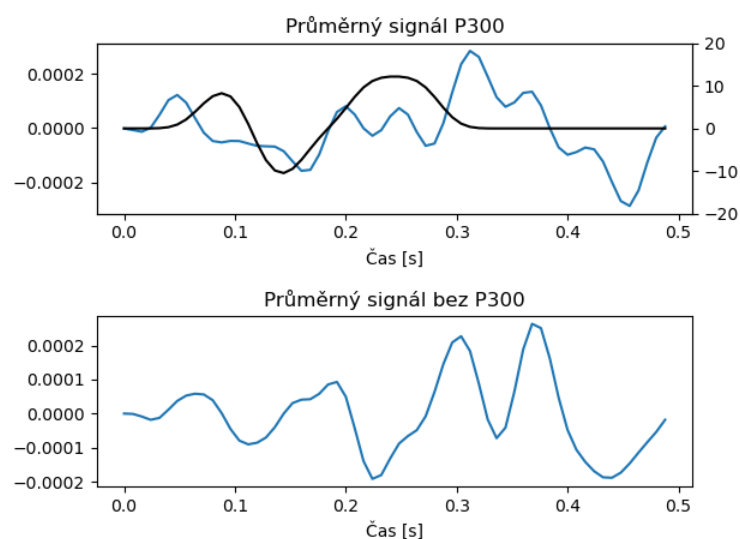


Obrázek 4.5: Vzorový signál vyprodukovaný ze 100 epoch umělého signálu, zobrazený spolu s ideální odezvou na c-VEP (černě), která tvoří základ umělého signálu

### 4.3 Výsledky systému P300

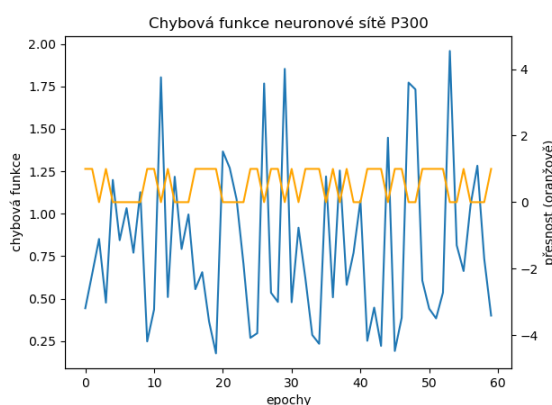
Systém využívající paradigmatu P300 byl testován v jednom sezení ve dvou pokusech. Data z pokusů byla uložena do dvou souborů dohromady obsahující 185 sekvencí signálu. Sekvence jsou označeny buď jako cílové, které obsahují P300, nebo necílové, které by P300 neměly obsahovat. Jak lze ovšem vidět na obrázku 4.6, tak průměrováním necílových signálů se objeví navýšení amplitudy zhruba ve stejném místě jako u cílové sekvence. Tento fenomén může být následkem podvědomé reakce na necílový sousední stimulus, který v tento moment probíkl. V průměru má ovšem cílový stimulus větší amplitudu v místě, kde by se dala očekávat vlna P300.

Chybovou funkci trénování neuronové sítě „one layer“ lze vidět na obrázku 4.7. Jak je vidět, chybová funkce opět nekonverguje, a síť není schopná se nic naučit. Přesnost sítě se vždy pohybuje v okolí poměru cílových sekvencí signálu na celkový počet sekvencí v datasetu a nelze tedy říci, že by výsledky byly lepší než náhodné. Podobných výsledků dosahovala druhá testovaná síť, kterou navrhl Gunsteren. Na datech syntetických, založených na přítomnosti vln P2, N3 a P3, který byl popsán v sekci 3.5, byly sítě opět schopné do-



Obrázek 4.6: Průměrný cílový (nahore) s naznačenou očekávanou podobou (černě) a necílový (dole) signál s pásovým filtrem od 1 do 20 Hz.

sáhnout přesných výsledků Chybová funkce trénování na syntetických datech je zobrazena na obrázku 4.8.

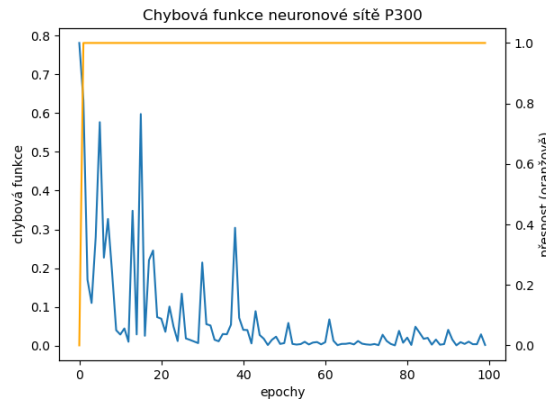


Obrázek 4.7: Chybová funkce trénovaného modelu „One layer“ (modře) a přesnost odhadu (oranžově)

## 4.4 Výsledky systému SSVEP

Systém SSVEP byl testován v jednom sezení ve dvou pokusech, které dohromady tvoří přes 90 sekvencí signálu. Data z pokusů byla uložena do souboru spolu s číslem značícím stimul, který byl právě cílen.

Na obrázku 4.9 je zobrazeno frekvenční spektrum některých stimulů, frekvence stimulů jsou označeny oranžovou vertikální linkou a stimul cílový vertikální linkou červenou. Je vidět, že frekvence stimulů nejsou dominantními frekvencemi v signále. Je velmi zvláštní,



Obrázek 4.8: Chybová funkce modelu „One layer“ (modře) trénovaného na syntetických datech a přesnost odhadu (oranžově)

že frekvence 8.57 Hz odpovídající druhému stimulu není ani v jednom vzorku signálu znatelně zastoupena. Také to vypadá, že při soustředění se na 1. a 2. stimul se zastoupení odpovídajících frekvencí 7.5 Hz a 8.57 Hz v signále nemění, což by byl zásadní problém.

Na obrázku 4.10 je znázorněna chybová funkce trénované neuronové sítě. Vypadá to, že neuronová síť prvních 30 epoch konverguje, ovšem za tímto bodem dále již není schopná se nic naučit, nebo se učí příliš pomalu. I když na první pohled graf vypadá, že neuronová síť dosáhla přesnosti okolo padesáti procent, tak přesnost na validačních datech se pohybovala mezi deseti a třiceti procenty a nelze tedy říct, že by se síť chovala lépe než náhodně.

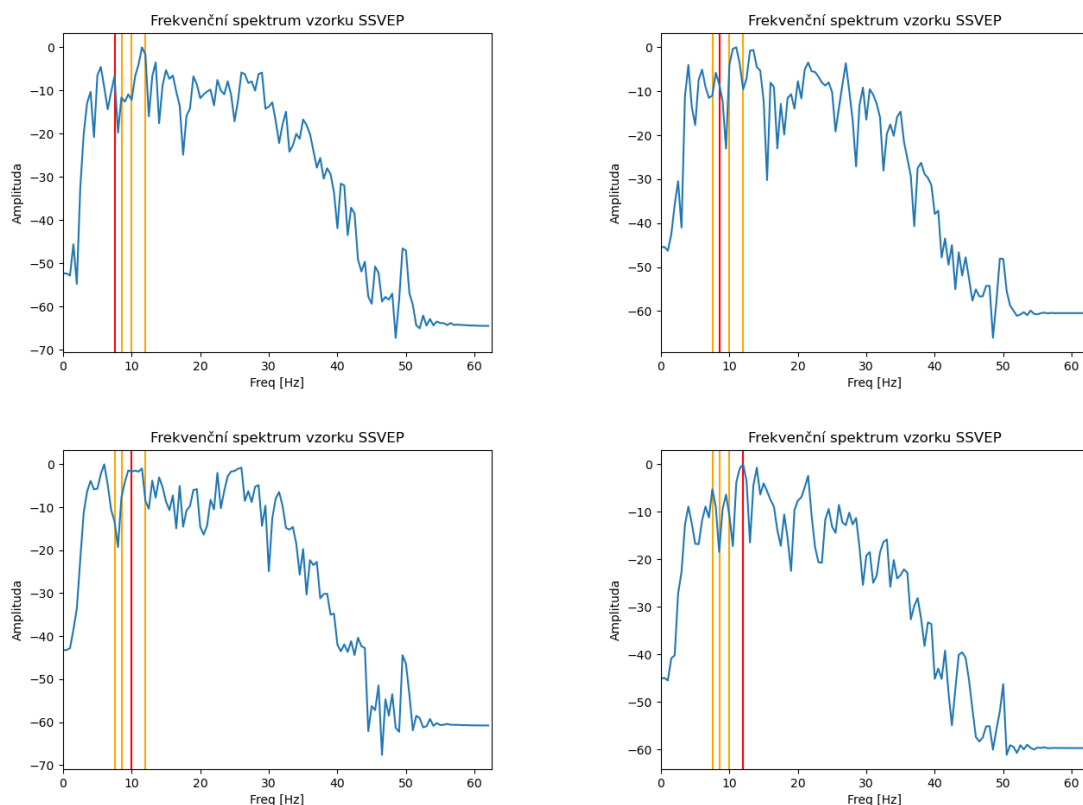
Opět bylo provedeno testování sítě na syntetických datech, jejichž generování je popsáno v sekci 3.5. Tato data obsahují jako svůj základ signál o frekvenci dané vybraným cílem. Po průchodu tohoto signálu předzpracováním lze jednoduše vyčíst přítomnost všech očekávaných příznaků, včetně přítomnosti frekvence 8.57 Hz, jak lze vyčíst z grafu 4.11. Takto je potvrzena použitelnost předzpracování. Neuronová síť pro SSVEP byla také schopna naučit se rozpoznávat jednotlivé cíle se 100% přesností a lze tedy předpokládat, že její architektura je použitelná pro klasifikaci SSVEP. Chybová funkce a přesnost trénování na syntetických datech je znázorněna v grafu 4.12.

## 4.5 Shrnutí výsledků a diskuze

Je zvláštní, že žádná z metod použitých pro klasifikaci nedosáhla relevantních výsledků na žádném paradigmatu vjemů. Tento problém naznačuje, že by chyba mohla být v některé společné části jako filtrace dat, způsob zachytávání signálu nebo dokonce v nevhodném subjektu.

Kdyby chyba byla ve filtraci, bylo by relativně jednoduché ji odstranit, problém ovšem nastává při odhalení takové chyby, které vyžaduje značné testování a pravděpodobně další a rozsáhlejší manuální prohlížení grafů signálů, tak jak bylo použito u testování metody P300.

Chyba ve způsobu zachytávání signálu byla již jedna odhalena. Chvilí po inicializaci EEG již sice EEG posílá data, ovšem tato data mají oproti skutečnosti menší naměřenou hodnotu napětí. Naměřené napětí s časem roste dokud nedosáhne skutečných hodnot. Tento problém byl vyřešen čekáním a zahazením prvních dvou sekund dat. Je možné, že jiný

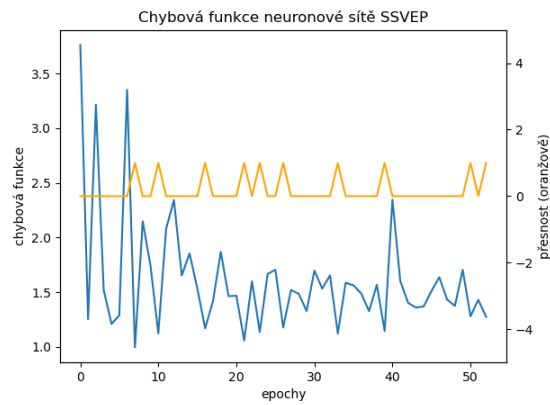


Obrázek 4.9: Frekvenční spektrum některých epoch. Na ose x frekvence v Hz. Vertikální linky naznačují frekvence stimulů, červená značí současný cíl.

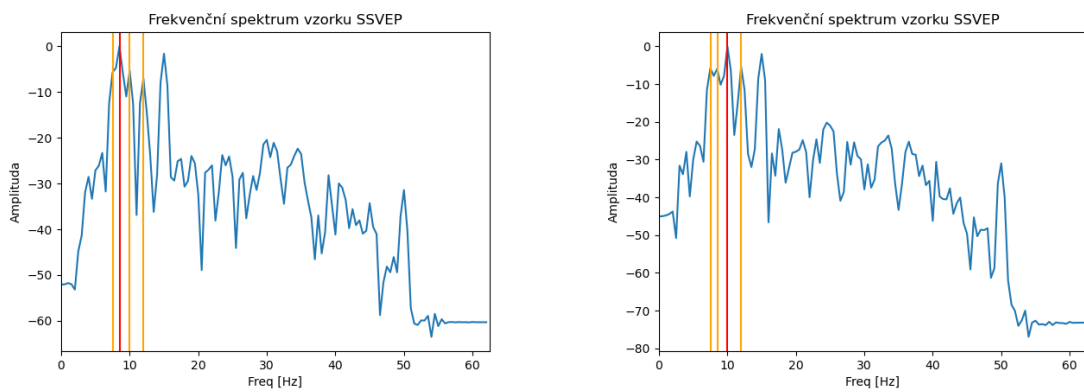
problém nastává při čtení dat, které může trvat i čtvrt sekundy, což je nezanedbatelná doba z hlediska vzorkovací frekvence 125 Hz. Otázkou je, jestli jsou data snímána během tohoto času uložena a vrácena při následujícím čtení, takto by totiž mohl vznikat problém s časováním některých stimulů.

Vážný problém může být kompatibilita subjektu. Jak bylo psáno výše, vzhledem k pandemické situaci a vyhlášení nouzového stavu byl systém testován pouze na jediném subjektu. Všechny citované studie testují systémy na více subjektech a každý subjekt je různě kompatibilní se systémem.

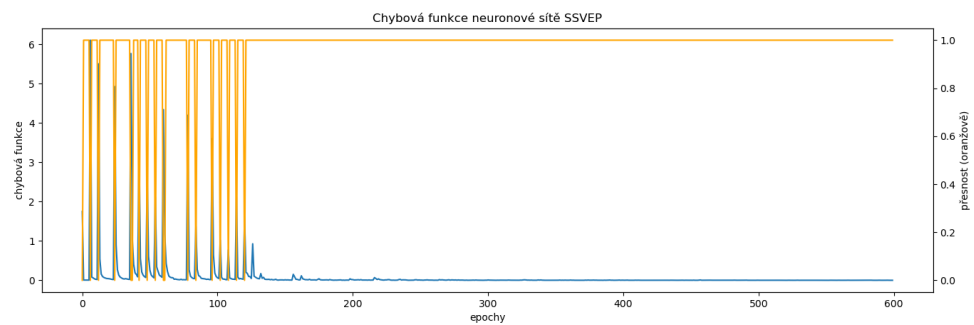
Je samozřejmě možné, že jednotlivé přístupy mají individuální problémy, které mohou nastávat například u prezentace stimulu, trénování sítí nebo při chybném výpočtu. Za účelem minimalizování rizika chybného výpočtu bylo odstoupeno od první iterace systému c-VEP, který počítal s celkovým posunutím a posunutím jednotlivých stimulů, bylo zapotřebí tedy někdy posunutí od sebe odečítat někdy přičítat, takto vznikala zmatek jehož následkem mohla být chyba. Tento přístup byl v druhé iteraci systému změněn na jediný posuv (posuv jednotlivých stimulů). A celkový posuv byl vždy nulován se začátkem sběru dat.



Obrázek 4.10: Chybová funkce neuronové sítě. Po 20 iteracích přestává konvergovat.



Obrázek 4.11: Frekvenční spektrum umělých signálů SSVEP s vyznačenými cíly.



Obrázek 4.12: Chybová funkce neuronové sítě pro klasifikaci SSVEP trénované na syntetických datech.

## Kapitola 5

# Závěr

V rámci této práce byla uvedena problematika brain-computer interface, která se zaměřuje na překlad mozkové aktivity na srozumitelné příkazy pro počítač. Účelem takovýchto aplikací je umožnění komunikace lidem trpícím ochrnutím končetin. Některé metodologie brain-computer interface se také využívají pro terapeutické účely. Cílem této práce bylo navrhnout a implementovat takovýto systém jako prostředek pro komunikaci s okolím pomocí převodu signálů na text.

V druhé kapitole byly následně rozebrány technologie brain-computer interface, elektroencefalografu a některé metody zpracování signálů. Elektroencefalograf je v práci využit ke snímání mozkové aktivity ze skalpu v podobě poměrně malých změn potenciálů vyvolaných bioelektrickými signály mozku. Velká část zpracování signálů byla zejména věnována strojovému učení se zaměřením na jeho podoblast hluboké učení. Především bylo popsáno použití konvolučních neuronových sítí, které se používají pro detekci lokálních příznaků v signálech i obrazu, a jejichž vstupem mohou být nezpracovaná data, nejčastěji v doméně časové nebo časově-frekvenční.

Třetí kapitola popisovala návrh systému a jeho podstatné části jako, jako grafické rozhraní, způsobující mozkovou odezvu, použité klasifikační algoritmy a struktury neuronových sítí a také detaily implementace a rozhodnutí k nim vedoucích. Došlo k implementaci čtyřech systémů. Systémy jsou založeny na rozdílných paradigmatech stimulu grafického rozhraní. K implementaci tří dalších systémů (systémů druhé iterace) a otestování více paradigmatů došlo z důvodu nefunkčnosti prvního implementovaného systému. Ovšem nakonec žádný ze systémů nebyl schopen spolehlivě překládat skutečná data.

Ve čtvrté kapitole došlo k otestování implementovaných systémů. Testovány byly části systému i systém jako celek na datech skutečných a na datech syntetických. Dvě části systému, které byly testovány, jsou předzpracování a klasifikace. Nejúčinnější předzpracování, ke kterým se došlo testováním, jsou uvedeny v sekci implementace. Na skutečných datech nebyl žádný ze systémů úspěšný, ovšem při použití syntetických vstupů neměly klasifikační algoritmy problém dosahovat vysokých přesností. Takto je potvrzena základní myšlenka klasifikačních algoritmů.

Problém je tedy pravděpodobně v datech, ovšem tento problém může mít více příčin. Jednou z možných příčin je nedostatek testovacích subjektů, vzhledem k pandemické situaci a vyhlášení nouzového stavu bylo možné testovat systém pouze na jediném subjektu. Další možnou příčinou by mohly být chybně tvořené vizuální stimuly, které by mohly špatně evokovat změny v mozkové aktivitě. Také je možné, že je chyba v synchronizaci dat a takto jsou evokované změny v průměru kompromitovány, ovšem tento problém by neměl nastávat u systému SSVEP.

Při dalším rozšiřování práce bych se zaměřil na experimenty s vizuálními stimuly používající jiné zařízení než obrazovku počítače, například MCU arduino s LED, dále bych experimentoval s jinými knihovnami jazyka a možná i více zařízeními EEG s rozdílnými parametry. Z hlediska rozšiřování programu by bylo možné rozšířit uživatelské rozhraní o systémovou nabídku a numerickou klávesnici. Také by nebylo od věci spojit systémy v jediný systém, za použití správných praktik objektově orientovaného programování, který by umožňoval volit mezi jednotlivými přístupy.

# Literatura

- [1] ABHANG, P. A., GAWALI, B. W. a MEHROTRA, S. C. Technological Basics of EEG Recording and Operation of Apparatus. In: *Introduction to EEG- and Speech-Based Emotion Recognition*. Elsevier, 2016, s. 19–50. DOI: 10.1016/b978-0-12-804490-2.00002-6. Dostupné z: <https://doi.org/10.1016/b978-0-12-804490-2.00002-6>.
- [2] ALVARADO GONZÁLEZ, M., FUENTES PINEDA, G. a CERVANTES OJEDA, J. A few filters are enough: Convolutional neural network for P300 detection. *Neurocomputing*. 2021, sv. 425, s. 37–52.
- [3] AMINAKA, D., MAKINO, S. a RUTKOWSKI, T. M. SVM classification study of code-modulated visual evoked potentials. In: *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, Prosinec 2015. DOI: 10.1109/apsipa.2015.7415435. Dostupné z: <https://doi.org/10.1109/apsipa.2015.7415435>.
- [4] BASHASHATI, A., FATOURECHI, M., WARD, R. K. a BIRCH, G. E. A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural Engineering*. IOP Publishing. mar 2007, sv. 4, č. 2, s. R32–R57. DOI: 10.1088/1741-2560/4/2/r03. Dostupné z: <https://doi.org/10.1088/1741-2560/4/2/r03>.
- [5] BIN, G., GAO, X., WANG, Y., HONG, B. a GAO, S. VEP-based brain-computer interfaces: time, frequency, and code modulations. *IEEE Computational Intelligence Magazine*. Institute of Electrical and Electronics Engineers (IEEE). listopad 2009, sv. 4, č. 4, s. 22–26. DOI: 10.1109/mci.2009.934562. Dostupné z: <https://doi.org/10.1109/mci.2009.934562>.
- [6] CHANG, M. H., BAEK, H. J., LEE, S. M. a PARK, K. S. An amplitude-modulated visual stimulation for reducing eye fatigue in SSVEP-based brain–computer interfaces. *Clinical Neurophysiology*. Elsevier BV. červenec 2014, sv. 125, č. 7, s. 1380–1391. DOI: 10.1016/j.clinph.2013.11.016. Dostupné z: <https://doi.org/10.1016/j.clinph.2013.11.016>.
- [7] GALLEGOS AYALA, G., FURDEA, A., TAKANO, K., RUF, C. A., FLOR, H. et al. Brain communication in a completely locked-in patient using bedside near-infrared spectroscopy. *Neurology*. Wolters Kluwer Health, Inc. on behalf of the American Academy of Neurology. 2014, sv. 82, č. 21, s. 1930–1932. DOI: 10.1212/WNL.0000000000000449. ISSN 0028-3878. Dostupné z: <https://n.neurology.org/content/82/21/1930>.



- [8] GONFALONIERI, A. *Deep Learning Algorithms and Brain-Computer Interfaces*. Listopad 2019. Dostupné z: <https://towardsdatascience.com/deep-learning-algorithms-and-brain-computer-interfaces-7608d0a6f01>.
- [9] HERRMANN, C. S. Human EEG responses to 1-100Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental Brain Research*. Springer Science and Business Media LLC. duben 2001, sv. 137, 3-4, s. 346–353. DOI: 10.1007/s002210100682. Dostupné z: <https://doi.org/10.1007/s002210100682>.
- [10] KHALAF, A., SYBELDON, M., SEJDIC, E. a AKCAKAYA, M. A brain-computer interface based on functional transcranial doppler ultrasound using wavelet transform and support vector machines. *Journal of Neuroscience Methods*. 2018, sv. 293, s. 174 – 182. DOI: <https://doi.org/10.1016/j.jneumeth.2017.10.003>. ISSN 0165-0270. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0165027017303515>.
- [11] KWAK, N.-S., MÜLLER, K.-R. a LEE, S.-W. A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PLOS ONE*. Public Library of Science (PLOS). únor 2017, sv. 12, č. 2, s. e0172578. DOI: 10.1371/journal.pone.0172578. Dostupné z: <https://doi.org/10.1371/journal.pone.0172578>.
- [12] LOTTE, F., BOUGRAIN, L., CICHOCKI, A., CLERC, M., CONGEDO, M. et al. A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of Neural Engineering*. IOP Publishing. duben 2018, sv. 15, č. 3, s. 031005. DOI: 10.1088/1741-2552/aab2f2. Dostupné z: <https://doi.org/10.1088/1741-2552/aab2f2>.
- [13] MÜLLER, K. R., KRAUEDAT, M., DORNHEGE, G., CURIO, G. a BLANKERTZ, B. Machine Learning and Applications for Brain-Computer Interfacing. In: *Human Interface and the Management of Information. Methods, Techniques and Tools in Information Design*. Springer Berlin Heidelberg, 2007, s. 705–714. DOI: 10.1007/978-3-540-73345-4\_80. Dostupné z: [https://doi.org/10.1007/978-3-540-73345-4\\_80](https://doi.org/10.1007/978-3-540-73345-4_80).
- [14] MUSE. A Deep DIve Into Brainwaves: Brainwave Frequencies Explained. Muse. 2018. Dostupné z: <https://choosemuse.com/blog/a-deep-dive-into-brainwaves-brainwave-frequencies-explained-2/>.
- [15] NAGEL, S. a SPÜLER, M. Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed Brain-Computer Interface. *PLOS ONE*. Public Library of Science (PLOS). říjen 2018, sv. 13, č. 10, s. e0206107. DOI: 10.1371/journal.pone.0206107. Dostupné z: <https://doi.org/10.1371/journal.pone.0206107>.
- [16] NAGEL, S. a SPÜLER, M. World’s fastest brain-computer interface: Combining EEG2Code with deep learning. *PLOS ONE*. Public Library of Science (PLOS). září 2019, sv. 14, č. 9, s. e0221909. DOI: 10.1371/journal.pone.0221909. Dostupné z: <https://doi.org/10.1371/journal.pone.0221909>.
- [17] NAKANISHI, M., WANG, Y., WANG, Y.-T., MITSUKURA, Y. a JUNG, T.-P. Generating Visual Flickers for Eliciting Robust Steady-State Visual Evoked Potentials at Flexible

Frequencies Using Monitor Refresh Rate. *PLoS ONE*. Public Library of Science (PLOS). červen 2014, sv. 9, č. 6, s. e99235. DOI: 10.1371/journal.pone.0099235. Dostupné z: <https://doi.org/10.1371/journal.pone.0099235>.

- [18] RASHID, M., SULAIMAN, N., MAJEED, A. P. P. A., MUSA, R. M., NASIR, A. F. A. et al. Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review. *Frontiers in Neurorobotics*. Frontiers Media SA. červen 2020, sv. 14. DOI: 10.3389/fnbot.2020.00025. Dostupné z: <https://doi.org/10.3389/fnbot.2020.00025>.
- [19] SHAN, H., LIU, Y. a STEFANOV, T. A Simple Convolutional Neural Network for Accurate P300 Detection and Character Spelling in Brain Computer Interface. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, červenec 2018. DOI: 10.24963/ijcai.2018/222. Dostupné z: <https://doi.org/10.24963/ijcai.2018/222>.
- [20] SOLON, A. J., LAWHERN, V. J., TOURYAN, J., MCDANIEL, J. R., RIES, A. J. et al. Decoding P300 Variability Using Convolutional Neural Networks. *Frontiers in Human Neuroscience*. Frontiers Media SA. červen 2019, sv. 13. DOI: 10.3389/fnhum.2019.00201. Dostupné z: <https://doi.org/10.3389/fnhum.2019.00201>.
- [21] SPÜLER, M., ROSENSTIEL, W. a BOGDAN, M. One Class SVM and Canonical Correlation Analysis increase performance in a c-VEP based Brain-Computer Interface (BCI). Unpublished. 2012. DOI: 10.13140/2.1.2186.7526. Dostupné z: <http://rgdoi.net/10.13140/2.1.2186.7526>.
- [22] THIELEN, J., MARSMAN, P., FARQUHAR, J. a DESAIN, P. From full calibration to zero training for a code-modulated visual evoked potentials brain computer interface. *Journal of Neural Engineering*. IOP Publishing. březen 2021. DOI: 10.1088/1741-2552/abecef. Dostupné z: <https://doi.org/10.1088/1741-2552/abecef>.
- [23] WOLPAW, J. R., BIRBAUMER, N., MCFARLAND, D. J., PFURTSCHELLER, G. a VAUGHAN, T. M. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*. Elsevier BV. jun 2002, sv. 113, č. 6, s. 767–791. DOI: 10.1016/s1388-2457(02)00057-3. Dostupné z: [https://doi.org/10.1016/s1388-2457\(02\)00057-3](https://doi.org/10.1016/s1388-2457(02)00057-3).
- [24] ZHANG, X., YAO, L., WANG, X., MONAGHAN, J. J. M., MCALPINE, D. et al. A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers. *Journal of Neural Engineering*. IOP Publishing. listopad 2020. DOI: 10.1088/1741-2552/abc902. Dostupné z: <https://doi.org/10.1088/1741-2552/abc902>.